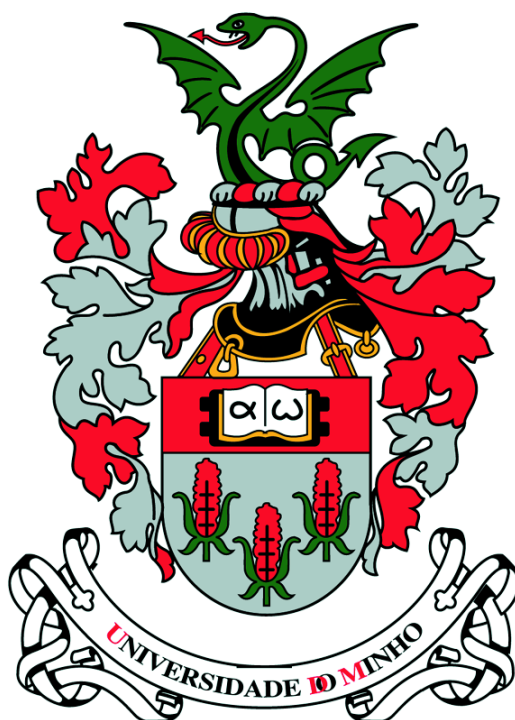


# Topic Maps: da Sintaxe à Semântica

Giovani Rubert Librelotto



*Dissertação submetida à Universidade do Minho para obtenção do grau de Doutor em Informática, elaborada sob a orientação do professor Doutor Pedro Rangel Henriques*

Departamento de Informática  
Escola de Engenharia  
Universidade do Minho  
Braga, 2005



## Resumo

Segundo a definição proposta em *Topic Maps Data Model* (Garshol and Moore, 2005), **Topic Maps** são *estruturas abstractas que podem codificar o conhecimento, conectando-o com recursos de informação relevantes*. Os **Topic Maps** permitem a estruturação da informação através de uma rede semântica composta por tópicos associados.

Actualmente, a maior parte dos **Topic Maps** são construídos manualmente. Este tipo de edição acarreta custos de ordem temporal e financeira, pois apesar de haver ferramentas propícias para a sua edição, as mesmas perdem eficiência quando o topic map atinge um número considerável de tópicos e associações. Acresce ainda o facto de que o utilizador tem dificuldade em verificar se a semântica do topic map condiz com o seu interesse.

Os **Topic Maps** possuem uma característica muito importante: a liberdade de representação de um universo de discurso, pois a definição de um tópico é muito ampla. Porém, esta liberdade pode representar um potencial perigo para a consistência do topic map. Para garantir essa consistência, um conjunto de condições contextuais (restrições semânticas) deve ser imposto ao topic map.

A norma **Topic Maps** não fornece nenhum tipo de mecanismo para validar a semântica de documentos topic maps de acordo com regras especificadas por utilizadores. Por isso, urge completar a norma com um suporte à definição de restrições contextuais e criar um mecanismo de validação automática. O principal contributo deste doutoramento é uma linguagem de restrições para topic maps, denominada **XTche**, e o respectivo processador. A linguagem **XTche** – baseada nos requisitos propostos recentemente em *TMCL (Topic Map Constraint Language)* (Nishikawa, Moore, and Bogachev, 2004) – permite a descrição da estrutura da rede semântica formada pelos tópicos e associações e a definição de restrições semânticas através de regras de esquema, regras contextuais e regras de existência.

Baseado nisto, decidiu-se pelo projecto e desenvolvimento de um ambiente que fosse capaz de extrair dados de recursos de informação e construir um topic map de acordo com uma especificação, validá-lo e permitir uma navegação conceptual sobre o conhecimento representado no topic map.

Resultou desta decisão o outro contributo deste doutoramento: o **Metamorphosis**, que é formado por um conjunto de linguagens de especificação e ferramentas que permitem criar uma interface para integração de informação oriunda de diversas fontes, através do uso de uma ontologia representada em **Topic Maps**. A partir da descrição das fontes heterogéneas de informação e da especificação da ontologia, o **Oveia** (um dos componentes do **Metamorphosis**) extrai automaticamente o respectivo topic map. Depois de guardado – num documento **XTM** (XML Topic Maps), ou numa base de dados – este topic map será validado sintáctica e semanticamente (face a um conjunto de restrições especificadas numa linguagem apropriada) pelo **Processador de XTche** (outro dos componentes). Por fim, a componente **Ulisses** gera uma interface Web para manipular o topic map extraído, a partir da descrição **XTM** válida. Estas componentes, algumas das quais com implementações alternativas ou mais que uma versão funcional, têm a particularidade de poderem ser usadas separadamente, tal foi comprovado nos casos de estudos realizados.



## Abstract

According to Topic Map Data Model (Garshol and Moore, 2005), **Topic Maps** are *abstract structures that can encode knowledge and connect this encoded knowledge to relevant information resources*. **Topic Maps** allow a domain knowledge representation in semantic networks, composed of topics and associations.

Nowadays, almost all topic maps are built by hand. This kind of edition is time consuming and has important financial costs. There are several tools for topic map edition but they have some limitations like the lack of a topic map semantic validator.

In order to cope with a broad range of scenarios, a topic is a very wide concept. On one hand, this makes **Topic Maps** a convenient model for knowledge representation; but on the other hand, this can also put in risk the topic map consistency. A set of semantic constraints must be imposed to the topic map in order to grant its consistency.

The **Topic Maps** standard does not provide language constructors to specify the semantics. So it is not possible to derive from the standard mechanisms to validate a topic maps against the contextual rules. Therefore it is necessary to improve the ISO 13250 standard adding a support for constraints definition enabling the creation of a processor for topic map automatic validation.

The main contribute of this thesis is a constraint language for topic maps called **XTche** and its processor. **XTche** language is TMCL-based (*Topic Map Constraint Language*) (Nishikawa, Moore, and Bogachev, 2004). This language allows to complement the description of the semantic network structure (composed of topic and associations) with schema, contextual, and existence constraints, thus defining the semantics of topic maps that should be preserved.

**Metamorphosis** – an environment that can extract data from information resources and build a topic map according to a specification, validate it, and generate a conceptual navigation over the topic map knowledge – is another contribution of this thesis. **Metamorphosis** – a **Topic Maps** oriented environment – generates conceptual navigators for heterogenous information resources providing the desired interoperability.

**Metamorphosis**' architecture is composed of: (1) **Oveia**, a processor that builds topic maps. Its core is a processor that extracts the topics instances from the information resources and builds a topic map. It reads and processes the **XSDS** and **XS4TM** specifications. The topic map generated by **Oveia** is stored as an **XTM** file or alternatively as a relational database following the **OntologyDB** approach; (2) **XTche** processor, that consumes the previous **XTM** file and validates the topic map according to a set of constraints defined in **XTche** language; (3) **Ulysses** processor, that produces a whole semantic website based on a valid topic map; this website is a set of pages that displays all the information concerned with topics and associations and provides a conceptual navigation over the semantic network (the topic map).



## Agradecimentos

*Quero ser teu amigo,  
Nem demais e nem de menos...  
Nem tão longe e nem tão perto.  
Na medida mais precisa que eu puder.  
Mas amar-te como próximo, sem medida...  
E ficar sempre em tua vida  
Da maneira mais discreta que eu souber.  
Sem tirar-te a liberdade,  
Sem jamais te sufocar,  
Sem forçar a tua vontade.  
Sem falar quando for a hora de calar  
E sem calar quando for a hora de falar.*

Fernando Pessoa

Ao Departamento de Informática, pelo acolhimento recebido ao longo destes 4 anos, em especial aos funcionários Paula Anjo, Goretti, Helena, Cristina, João, Carla e Jaime.

Ao grupo de Especificação e Processamento de Linguagens, especialmente ao Zé João, Alberto, Jorge Gustavo, Sandra Lopes, Fernanda, Paulo Matos e Alda Lopes, por formarem um grupo que tem a cara de uma família.

Uma menção distinta à Marta Jacinto, pelas conversas produtivas ao longo da duração da sua dissertação de mestrado. O desenvolvimento de XCSL trouxe mais que algumas publicações: criou uma boa amizade.

Alunos da disciplina de Processamento Estruturado de Documentos Fernando Jorge Laúndos Ferreira da Costa e Mário André de Castro Vila Cova, pelo empenho no desenvolvimento do Ulisses.

Ao Weber Souza pelo convívio virtual que deu origem ao Oveia. A orientação do seu trabalho de conclusão de curso na Universidade do Estado da Bahia rendeu bons frutos.

Ao José Carlos Ramalho, pela receptividade desde o momento da chegada em Portugal, naquele dia 27 de Setembro de 2001, no aeroporto Sá Carneiro. Uma pessoa totalmente informada da área onde actua, sempre com novas ideias para dar sequência aos projectos. Toda esta dissertação teve sua participação na idealização e concretização, por isso, és uma pessoa fundamental pela obra que aqui se apresenta. Sou muito grato a você.

Ao CNPq, pelo apoio financeiro ao longo deste doutoramento. Em especial ao professor Doutor Rafael Lins e aos funcionários Katiane Regia Menezes, Eli Ribeiro da Silva e José Airton de Souza, que sempre estiveram à disposição nos momentos de dúvidas burocráticas.

Ao professor Doutor Simão Sirineo Toscani, um agradecimento particular: se esse doutoramento está sendo finalizado agora, é porque começou dia 03 de Agosto de 2000, quando me indicaste o professor Pedro Henriques como possível orientador. Fizeste muito por mim orientando-me durante a faculdade e o mestrado, sendo mais que um orientador; foste sempre um amigo. E nem o senhor sabe o quão bem me fizeste, ao apresentar-me meu actual orientador. Ter o senhor no juri de doutoramento é uma satisfação e uma expressão de gratidão.

Aos grandes amigos portugueses, que levarei para sempre comigo: David Costa, com nossos planos de viagem que ainda não foram colocados em prática (mas não tá morto quem peleia!), e o Pedro Gabriel, um companheiro de gabinete ímpar, tão boa gente que até chego a relevar o seu péssimo gosto musical. Espero vocês no Brasil, gajos.

Ao Ricardo Martins, pelos 2 anos de convivência em solo português. Este tempo provou que pessoas diferentes podem conviver muito bem e se tornar bons amigos.

À Giovana, apesar de ficar pouco no DI semanalmente, soube sempre trazer aquele ar feminino para o meio desse bando de machos que vieram do Brasil pra Braga. Ainda temos que reviver aquelas noites de War com o Gustavo e o Ricardo.

À comunidade baiana do DI: o Fabio, aquele cara sensato e correto, uma excelente e motivante companhia, principalmente para aguentar os finais de semana na universidade; o Tiago, o cara com a voz de dominado no telefone com a Tati, sempre pronto pra um papo cabeça com a galera; e o Vitor, chegou por último mas soube trazer um novo astral pro apartamento, o mais irado dos brazucas.

À família “paraúcha” Ronnie, Ana e Pedrinho Miguel, pela convivência ao longo do tempo que moramos juntos, por trazer o lado família, algo que faltava por aqui.

Ao Alf, pela sua personalidade singular. Uma pessoa que parece ser fechada e sisuda, porém sabe se expressar como poucos. Devia estar no *Guinness* não por seu casamento relâmpago (né, Maria?), mas sim pelo seu potencial de trabalho. Esse sim é um exemplo a seguir.

O Gustavo merece uma citação à parte. O tempo total desse doutoramento estivemos lado a lado, seja morando no mesmo apartamento, seja no gabinete (quando você vai trabalhar, claro!). Um cara sempre muito bem informado, principalmente no quesito fotografia digital e turismo. Um dia ainda descobrirás que estás no curso errado... Você é o mais oveia dos oveias. ;)

Ao Marco, sem palavras... Faculdade, mestrado e doutorado juntos. Você é meu ídolo, tche, não foi a toa que abri as portas da UMinho para você. Sabia que você merecia mais que ninguém estar numa instituição como essa. Sei que além de mim, o professor Luis Barbosa, seu orientador, também lhe agradece.

Aos amigos da minha grande pequena Fortaleza dos Valos: os cumpadres Zando e Adri, por aquele cafofo acolhedor em Cruz Alta e a ótima companhia; Fernando e Ângela, pela amizade e nossa praia, a qual vai acontecer mais e mais vezes; os cumpadres Rafão e Roberta, pela honra de apadrinhar vosso casamento; e a todos os demais, que foram importantes em qualquer fase desta jornada.

À minha família em geral, tios, tias, primos e primas. Citar o nome de todos os quase 100 membros seria escrever como outra dissertação. Para representar todos, cito meu avô Américo, falecido em 2001, justamente quando eu estava visitando minha família pela primeira vez depois de ter vindo à Braga. Gostaria de estar lhe entregando esse título pessoalmente. Mas estejas onde estiveres, saberás que tem em mim um neto muito orgulhoso de ser um descendente seu. O senhor faz muita falta.

Ao meu anjo, Daniele, agradeço pela compreensão nestes últimos 6 meses quando eu não podia dar a atenção que você merecia. Obrigado por tornar mágico cada fim de noite, mesmo estando longe de mim.

Os cumpadres Jairo, Joice e a linda Sassá merecem destaque. Já tenho meu lugarzinho naquela casa. Obrigado não só pelos jantares e bons momentos em todas as minhas



viagens ao Brasil, mas principalmente por quem vocês são. Vocês foram fundamentais nesse período em terras lusitanas e faço questão de manter essa grande amizade para sempre.

Aos meus gurus Guigo e Tiago, um reconhecimento. Quando viram que eu estava inclinado a não encarar esse doutorado, souberam como me motivar e fazer eu ver que estaria cometendo o maior erro da minha vida. Guigo, nossas eternas conversas pela internet, quase que diárias, sempre me fazendo sentir um pouquinho mais em casa. Não tens ideia do quanto essas conversas foram importantes para mim. Tiago, mesmo com 4 anos a menos em idade, sempre tiveste 4 anos a mais em maturidade. Conversar contigo sempre me trouxe algo novo, uma lição. Sua visita aqui em Portugal foi um momento que para sempre ficará marcado. Ainda temos aquelas 3 garrafas de Vinho do Porto para degustar, uma por década. Meus bruxos, não nascemos da mesma mãe, mas vocês sabem que são irmãos pra mim.

E falando em irmãos, como não citar a Bibi e o Dodo... Esses sim, irmãos de sangue que pude acompanhar desde o nascimento. Hoje estão fazendo faculdade, mas verei sempre vocês como aquelas crianças que vieram trazer mais felicidades ao nosso lar. A Bibi, com sua personalidade forte, sempre brigando pelos seus direitos, sem perder a ternura. O Dodo, com seu jeito tímido e um potencial e tanto. Muito me orgulho de vocês, maninhos... Meu mano Ogi, temos uma vida juntos. E por não ser suficiente o tempo que estivemos juntos no Brasil, você acabou vindo para cá fazer seu mestrado. Mais que isso, acabou trazendo aquilo que mais me faltava: a minha própria família. Ter você aqui nos aproximou ainda mais, solidificando essa fraternidade que é uma bela amizade. Obrigado por ter vindo.

Para finalizar, meus pais... Nem Fernando Pessoa (com todo o respeito) teria talento com as palavras para expressar o quanto sou grato por vocês, pela maneira como conduzem aquele lar, pela forma como educaram seus filhos, pela dedicação a todos nós, sempre abrindo mão de suas prioridades, para nos oferecer do bom e do melhor. Quando eu estava em dúvida se viria para Portugal, surpreendentemente vocês foram os primeiros a me apoiar. Quando pensava que vocês estavam satisfeitos por eu ser um professor universitário em Cruz Alta, morando perto de todos, mostraram que eu não podia me acomodar com aquilo. Sois os responsáveis por esta conquista. Na verdade, esse título deveria ser dado a vocês.

E falando em pai, vou ter que parar de chamar o professor Pedro por professor. Desde o princípio soubeste como me fazer sentir em casa. Naqueles telefonemas quase diários, sempre estavas preocupado em saber como eu estava, se estava tudo bem; somente depois disso, vinha o trabalho. No fundo, nunca me viste como um orientando; sempre fui tratado como um filho. Estes quatro anos foram memoráveis, em muito graças ao senhor. Sou uma pessoa muito enraizada em minha cidade, família e amigos, mas graças principalmente ao senhor, eu tive motivação para percorrer todas as etapas deste doutoramento. Espero ter sido pelo menos metade do que o senhor foi para mim. Pois nem sob encomenda eu encontraria uma pessoa, em outro país, em outro mundo, como o senhor. Muito obrigado, do fundo do meu coração.



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Restrições em Topic Maps . . . . .	3
1.2	A História deste doutoramento . . . . .	4
1.3	A Tese e os Contributos . . . . .	6
1.4	Estrutura da Dissertação . . . . .	6
<b>I</b>	<b>Topic Maps: Definição e Representação</b>	<b>9</b>
<b>2</b>	<b>Representação de Conhecimento e Semantic Web</b>	<b>11</b>
2.1	Dado, Informação e Conhecimento . . . . .	12
2.2	Estruturando o Conhecimento . . . . .	13
2.2.1	Estruturas de Facetas . . . . .	14
2.2.2	Dicionários . . . . .	15
2.2.3	Índices . . . . .	16
2.2.4	Taxonomias . . . . .	16
2.2.5	Thesauri . . . . .	18
2.2.6	Ontologias . . . . .	20
2.2.7	Conexão entre os modelos . . . . .	24
2.3	Semantic Web . . . . .	25
2.3.1	Arquitectura da Semantic Web . . . . .	26
2.4	Linguagens para a Semantic Web . . . . .	28
2.4.1	Resource Description Framework – RDF . . . . .	28
2.4.2	RDF Schema – RDF(S) . . . . .	31
2.4.3	Topic Maps – TM . . . . .	33
2.4.4	XML-based Ontology Exchange Language – XOL . . . . .	34
2.4.5	Simple HTML Ontology Extensions – SHOE . . . . .	36
2.4.6	Web Ontology Language – OWL . . . . .	37
2.4.7	Comparativo entre as linguagens . . . . .	40
2.5	Sumário do capítulo . . . . .	41

<b>3</b>	<b>Topic Maps</b>	<b>43</b>
3.1	A norma ISO 13250 Topic Maps . . . . .	45
3.1.1	XML Topic Maps (XTM) . . . . .	46
3.1.2	TAO - Tópicos, Associações e Ocorrências . . . . .	47
3.1.3	Nomes de Tópicos ( <i>Topic Names</i> ) . . . . .	55
3.1.4	Nomes Variantes . . . . .	56
3.1.5	Contexto ( <i>Scope</i> ) . . . . .	57
3.2	Fusão de Topic Maps . . . . .	59
3.3	Identificador e Indicador de Tema: definição da identidade de um tema . .	61
3.4	Linguagens concretas para Topic Maps . . . . .	63
3.4.1	A sintaxe HyTM . . . . .	63
3.4.2	A sintaxe AsTMa= . . . . .	65
3.4.3	A sintaxe LTM . . . . .	65
3.5	Justificativa da escolha de Topic Maps em vez de RDF . . . . .	67
3.5.1	Diferenças históricas . . . . .	67
3.5.2	Diferenças à primeira vista . . . . .	68
3.5.3	Entendendo as normas . . . . .	68
3.5.4	Comparando as normas . . . . .	69
3.5.5	Por que usar Topic Maps como norma? . . . . .	74
3.6	Sumário sobre Topic Maps . . . . .	75
<b>4</b>	<b>Ontologias e Topic Maps</b>	<b>77</b>
4.1	Ontologia relacionada com modelos de dados . . . . .	78
4.2	Como construir uma Ontologia . . . . .	79
4.3	Ontologia em Topic Maps . . . . .	80
4.4	Topic Maps dirigidos por Ontologia . . . . .	82
4.4.1	Como Ontologias se relacionam com Topic Maps . . . . .	82
4.4.2	Vantagens da abordagem <i>TMdO</i> . . . . .	82
4.4.3	As tendências para a abordagem <i>TMdO</i> . . . . .	84
4.5	Sumário sobre Ontologia e Topic Maps . . . . .	84
<b>5</b>	<b>Validação Semântica em Topic Maps</b>	<b>87</b>
5.1	Restrições . . . . .	87
5.2	A necessidade de validação em Topic Maps . . . . .	88
5.3	O que já se pode restringir? . . . . .	90
5.3.1	Com a norma Topic Maps . . . . .	90
5.3.2	Em XML Topic Maps . . . . .	91
5.4	ISO 19756: Topic Map Constraint Language – TMCL . . . . .	93
5.4.1	Restrições em Associações . . . . .	94
5.4.2	Restrições em Tópicos . . . . .	95

5.4.3	Restrições em Ocorrências . . . . .	96
5.4.4	Acréscimos à lista de Requisitos de TMCL . . . . .	97
5.5	O processo de validação de Topic Maps . . . . .	99
5.6	Sumário sobre Validação Semântica em Topic Maps . . . . .	100
<b>II</b>	<b>Topic Maps: Um Ambiente Processamento</b>	<b>101</b>
<b>6</b>	<b>Metamorphosis</b>	<b>103</b>
6.1	Interoperabilidade em Sistemas de Informação . . . . .	104
6.2	Introdução ao Metamorphosis . . . . .	106
6.3	Arquitetura do Metamorphosis . . . . .	107
6.4	Aplicações do Metamorphosis . . . . .	109
6.5	Sumário sobre o Metamorphosis . . . . .	110
<b>7</b>	<b>TM-Builder</b>	<b>113</b>
7.1	OntBuild: O Extractor Abstracto de Ontologias . . . . .	114
7.1.1	Uma linguagem para representação de ontologias no OntBuild . . . . .	115
7.2	TM-Builder: Um extractor de Topic Maps . . . . .	115
7.3	XSTM: Uma linguagem XML para especificar Extractores de Topic Maps . . . . .	116
7.3.1	Especificação de Tipos de Tópicos em XSTM . . . . .	117
7.3.2	Especificação de extracção de Tópicos em XSTM . . . . .	120
7.3.3	Especificação de Tipos de Associações em XSTM . . . . .	122
7.3.4	Especificação de extracção de Associações em XSTM . . . . .	123
7.4	Trabalhos Relacionados com o TM-Builder . . . . .	128
7.5	Sumário sobre TM-Builder . . . . .	129
<b>8</b>	<b>Oveia</b>	<b>131</b>
8.1	Arquitetura do Oveia . . . . .	132
8.2	XSDS: Especificação das Fontes de Dados . . . . .	133
8.2.1	Datasources: definição dos recursos de informação . . . . .	134
8.2.2	Datasets: definição da extracção dos conjuntos de dados . . . . .	134
8.2.3	Sintaxe de XSDS . . . . .	135
8.2.4	Exemplo de Especificação em XSDS . . . . .	136
8.3	Extractor de Datasets . . . . .	138
8.4	XS4TM: Uma linguagem XML para especificar a extracção de Topic Maps . . . . .	139
8.4.1	A sintaxe de XS4TM . . . . .	140
8.4.2	Exemplo de Especificação em XS4TM . . . . .	142
8.5	Processador XS4TM . . . . .	143
8.6	Armazenamento de Topic Maps em base de dados relacionais . . . . .	144
8.6.1	Armazenamento de XML em Base de Dados . . . . .	144

8.6.2	Solução Adoptada: BD Ontologia . . . . .	146
8.7	Trabalhos Relacionados com o Oveia . . . . .	148
8.8	Sumário sobre o Oveia . . . . .	149
<b>9</b>	<b>XTche</b>	<b>151</b>
9.1	Classificando as restrições sobre Topic Maps . . . . .	153
9.2	Estendendo XML Schema para XTche . . . . .	155
9.2.1	O esquema básico de uma especificação XTche . . . . .	156
9.2.2	Elementos e atributos específicos da linguagem XTche . . . . .	158
9.3	Restringindo Topic Maps em XTche . . . . .	162
9.3.1	Especificação de Restrições de Esquema . . . . .	162
9.3.2	Especificação de Restrições Contextuais . . . . .	164
9.3.3	Especificação de Restrições de Existência . . . . .	166
9.4	Estruturando Topic Maps com XTche . . . . .	167
9.5	Processando XTche . . . . .	172
9.5.1	TM-Validator . . . . .	173
9.5.2	Validação de Especificações XTche . . . . .	174
9.6	Trabalhos Relacionados com XTche . . . . .	175
9.7	Sumário sobre XTche . . . . .	176
<b>10</b>	<b>Ulisses</b>	<b>179</b>
10.1	Visualização de Topic Maps . . . . .	180
10.1.1	Visualização de Tópicos e Associações . . . . .	180
10.1.2	Filtragem em Topic Maps . . . . .	181
10.1.3	Abordagem escolhida . . . . .	181
10.2	Ulisses – o navegador . . . . .	183
10.2.1	Ulisses I – Navegação em Topic Maps baseada em transformações XSL	184
10.2.2	Ulisses II – Navegação em Topic Maps com transformação em tempo de execução . . . . .	188
10.2.3	Ulisses III – Navegação em Topic Maps armazenados na BD Ontologia	189
10.3	Trabalhos Relacionados com o Ulisses . . . . .	190
10.4	Sumário sobre o Ulisses . . . . .	192
<b>III</b>	<b>Topic Maps: Casos de Estudo</b>	<b>195</b>
<b>11</b>	<b>O Ulisses aplicado à navegação em Mapas de Conceitos</b>	<b>197</b>
11.1	Mapas de Conceitos . . . . .	198
11.2	Mapas de Conceitos em Topic Maps . . . . .	200
11.3	Um Topic Map para representar um Mapa de Conceitos . . . . .	202
11.3.1	Escolha da raiz (ou símbolo inicial) do Topic Map . . . . .	202

11.3.2	Definição dos tópicos correspondentes aos conceitos do MC . . . . .	202
11.3.3	Definição das associações correspondentes às relações do MC . . . . .	203
11.3.4	Definição das ocorrências de cada tópico . . . . .	205
11.4	Geração do Navegador para o Mapa de Conceitos . . . . .	206
11.5	Sumário do caso de estudo do IIPCS . . . . .	208
<b>12</b>	<b>O TM-Builder e o Oveia para a extracção do mapa do site do DI-UM</b>	<b>211</b>
12.1	Definição de uma ontologia para o DI-UM . . . . .	213
12.1.1	Os conceitos no domínio do DI-UM . . . . .	213
12.1.2	Uma taxonomia para o DI-UM . . . . .	214
12.1.3	Relacionamentos entre os conceitos da taxonomia . . . . .	214
12.2	TM-Builder aplicado ao DI-UM . . . . .	215
12.2.1	<i>Dump</i> da base de dados para um documento XML . . . . .	216
12.2.2	Especificação em XSTM . . . . .	217
12.2.3	Processamento e resultados do TM-Builder aplicado ao DI-UM . . . . .	222
12.3	Oveia aplicado ao DI-UM . . . . .	223
12.3.1	Especificação em XSDS . . . . .	224
12.3.2	Especificação em XS4TM . . . . .	225
12.3.3	Processamento e resultados do Oveia aplicado ao DI-UM . . . . .	228
12.4	Sumário do caso de estudo do DI-UM . . . . .	229
<b>13</b>	<b>A linguagem XTche numa aplicação de E-Commerce</b>	<b>233</b>
13.1	Modelo de dados da <i>E-Sell</i> . . . . .	234
13.1.1	A entidade <i>produto</i> . . . . .	235
13.1.2	A entidade <i>cliente</i> . . . . .	236
13.1.3	A entidade <i>ordem de compra</i> . . . . .	237
13.1.4	A relação <i>conter</i> . . . . .	238
13.1.5	A relação <i>comprar</i> . . . . .	240
13.2	Restrições sobre o topic map da <i>E-Sell</i> . . . . .	241
13.3	Restrições XTche sobre o topic map da <i>E-Sell</i> . . . . .	242
13.3.1	Restrições sobre a entidade <i>cliente</i> . . . . .	242
13.3.2	Restrições sobre a entidade <i>produto</i> . . . . .	244
13.3.3	Restrições sobre a entidade <i>ordem de compra</i> . . . . .	245
13.3.4	Restrições sobre o relacionamento <i>comprar</i> . . . . .	245
13.3.5	Restrições sobre o relacionamento <i>conter</i> . . . . .	246
13.3.6	Restrições Contextuais . . . . .	247
13.4	Processamento da especificação XTche e validação do topic map da <i>E-Sell</i> . . . . .	248
13.5	Sumário do caso de estudo da <i>E-Sell Corporation</i> . . . . .	250
<b>14</b>	<b>O Metamorphosis aplicado ao Museu da Emigração</b>	<b>253</b>

14.1	Sistema de Informação do Museu da Emigração . . . . .	255
14.1.1	Notas Biográficas contidas em Almanaque . . . . .	255
14.1.2	Diários de Viagem . . . . .	257
14.1.3	Registos de Passaportes . . . . .	258
14.2	Passos para a Integração de Fontes Heterogéneas . . . . .	260
14.3	Construção do topic map pelo Oveia . . . . .	261
14.3.1	Especificação das fontes em XSDS . . . . .	261
14.3.2	Definição da ontologia do universo de discurso . . . . .	264
14.3.3	Especificação da Ontologia em XS4TM . . . . .	265
14.3.4	Instanciação da Ontologia em XS4TM . . . . .	267
14.3.5	Extracção do topic map com o Oveia . . . . .	269
14.4	Validação do topic map com XTche . . . . .	272
14.5	Navegação no topic map com Ulisses . . . . .	275
14.6	Sumário do caso de estudo do Museu da Emigração . . . . .	277
<b>15</b>	<b>Conclusão</b>	<b>279</b>
15.1	Contributos . . . . .	281
15.2	Trabalhos Futuros . . . . .	281
<b>A</b>	<b>Regras para a Validação de Especificações XTche</b>	<b>299</b>
<b>B</b>	<b>Comparando XTche, AsTMa! e OSL</b>	<b>301</b>
B.1	A linguagem de restrições AsTMa! . . . . .	301
B.1.1	<i>Maplet Patterns</i> . . . . .	301
B.2	A linguagem de restrições OSL . . . . .	303
B.3	Comparando XTche, AsTMa! e OSL . . . . .	304
B.3.1	Condições estruturais rígidas . . . . .	304
B.3.2	Condições estruturais não-rígidas . . . . .	306
B.3.3	Restrições de Existência . . . . .	307
B.3.4	Restrições Condicionais . . . . .	307
B.3.5	Restrições Booleanas de Negação . . . . .	309
B.3.6	Restrições Booleanas – “E” e “OU” . . . . .	310



# Lista de Figuras

2.1	A linha contínua da interpretação. Fonte: Jack Park (2003).	12
2.2	Uma taxonomia para o planeta Terra.	17
2.3	Um thesaurus para o planeta Terra.	19
2.4	Uma ontologia para o planeta Terra.	22
2.5	Relacionamento entre ontologia, thesaurus, taxonomia, índice e dicionário	24
2.6	Arquitetura da <i>Semantic Web</i> . Fonte: Tim Berners-Lee (2000).	27
2.7	Esquema de representação da RDF em grafo	28
2.8	Diagrama de nós e arcos de RDF	29
2.9	Grafo RDF exemplo	30
2.10	Um exemplo envolvendo RDF e RDF Schema	32
2.11	Exemplo de aplicação de RDF Schema.	33
2.12	Exemplo de ontologia em XOL.	35
3.1	Esquema da sintaxe XTM	48
3.2	Ilustração Simples da Norma Topic Maps	49
3.3	Tópicos	49
3.4	Tipos de Tópicos	50
3.5	Super-classes e sub-classes	51
3.6	Ocorrências de Tópicos	53
3.7	Associações de Tópicos	54
3.8	Nomes de tópicos em diversos contextos	58
3.9	Dois topic maps antes da união. Fonte: The Topic Maps Handbook (2003).	59
3.10	Dois topic maps após da união. Fonte: The Topic Maps Handbook (2003).	60
3.11	Temas não-endereçáveis. Fonte: The Topic Maps Handbook (2003).	62
3.12	Identidade de dois topic maps. Fonte: The Topic Maps Handbook (2003).	62
3.13	As famílias das duas normas. Fonte: Living with topic maps and RDF (2003).	68
3.14	Reificando uma Associação. Fonte: Lars Marius Garshol (2003).	73
3.15	Reificando uma Declaração. Fonte: Lars Marius Garshol (2003).	73
4.1	Um topic map para o conceito <i>Dinamização Científica</i>	80

5.1	Representação de um topic map na sintaxe XTM e sua visão em forma de grafo. . . . .	92
6.1	Perspectivas em interoperabilidade de sistemas de informação. . . . .	105
6.2	Arquitectura simplificada do <i>Metamorphosis</i> . . . . .	107
6.3	Arquitectura completa do <i>Metamorphosis</i> . . . . .	108
7.1	A arquitectura do sistema <i>OntBuild</i> . . . . .	115
7.2	Arquitectura do <i>TM-Builder</i> . . . . .	116
7.3	Esquema do elemento raiz de <i>XSTM</i> . . . . .	117
7.4	Estrutura de um tipo de tópico em <i>XSTM</i> . . . . .	118
7.5	XML-Schema do <i>XATA</i> . . . . .	118
7.6	Estrutura da definição de extracção de tópicos em <i>XSTM</i> . . . . .	120
7.7	Estrutura de um tipo de associação em <i>XSTM</i> . . . . .	122
7.8	Estrutura da definição de extracção de associações em <i>XSTM</i> . . . . .	123
7.9	Estrutura da definição de extracção de associações <i>one2one</i> em <i>XSTM</i> . . .	124
7.10	Estrutura da definição de extracção de associações <i>M2N</i> em <i>XSTM</i> . . . .	127
8.1	Arquitectura do <i>Oveia</i> . . . . .	132
8.2	Relação entre <i>datasources</i> e <i>datasets</i> em <i>XSDS</i> . . . . .	138
8.3	Exemplo de funcionamento do <i>Extractor de Datasets</i> . . . . .	139
8.4	Relação entre <i>XSDS</i> e <i>XS4TM</i> . . . . .	141
8.5	Relação entre <i>XSDS</i> e <i>XS4TM</i> . . . . .	143
8.6	Trecho do Modelo ER do BD <i>Ontologia</i> . . . . .	147
9.1	Estrutura inicial de uma especificação <i>XTche</i> . . . . .	156
9.2	Estrutura do tipo de tópico <i>Artigo</i> com atributos <i>XTche</i> . . . . .	159
9.3	Regras contextuais para os tópicos <i>inglês</i> , <i>português</i> e <i>URL-PDF</i> . . . . .	160
9.4	Regras para verificação de existência para o tópico <i>XML2004-XTche</i> . . . .	161
9.5	Definição do tipo de tópico <i>cliente</i> com exactas três ocorrências . . . . .	161
9.6	Uma especificação <i>XTche</i> para definir a estrutura do tópico <i>country</i> . . . .	163
9.7	Especificação <i>XTche</i> para definir a estrutura da associação <i>comprar</i> . . . .	163
9.8	Restrição contextual determinando que o tópico <i>perfil</i> é, exclusivamente, um tipo de ocorrência . . . . .	165
9.9	Restrição contextual determinando que o tópico <i>artigo</i> não pode ser tipo de tópico, nem contexto de associação . . . . .	166
9.10	Exemplo de especificação de restrição de existência rígida . . . . .	166
9.11	Exemplo de especificação de restrição de existência não-rígida . . . . .	167
9.12	Gerador de Esqueleto XTM a partir de especificações <i>XTche</i> . . . . .	167
9.13	Especificação <i>XTche</i> para o tipo de tópico <i>cliente</i> . . . . .	169
9.14	Especificação <i>XTche</i> para o tipo de associação <i>conter</i> . . . . .	171

9.15	Arquitetura de XTche . . . . .	173
9.16	Processador para a Validação de Especificações XTche . . . . .	175
10.1	Visualização inicial de um topic map no <i>Ulisses</i> . . . . .	184
10.2	Visualização da ontologia de um topic map no <i>Ulisses</i> . . . . .	186
10.3	Visualização de um tópico no <i>Ulisses</i> . . . . .	187
10.4	Visualização de uma associação no <i>Ulisses</i> . . . . .	188
10.5	Visualização de um tópico no Omnigator . . . . .	190
10.6	Visualização de um tópico no Omnigator . . . . .	192
11.1	Exemplo de um Mapa de Conceitos simples . . . . .	199
11.2	Um Mapa de Conceitos referente à disciplina de IipCS . . . . .	200
11.3	Página do inicial do IipCS gerada pelo <i>Ulisses</i> . . . . .	206
11.4	Página referente ao conceito <i>Unidades</i> . . . . .	207
11.5	Página referente ao conceito <i>Processamento Numérico</i> . . . . .	207
11.6	Página referente ao conceito <i>Computador</i> . . . . .	208
11.7	Página referente ao conceito <i>Hardware</i> . . . . .	209
12.1	Arquitetura do website do DI-UM . . . . .	212
12.2	Uma taxonomia para o DI-UM . . . . .	214
12.3	A visão gráfica de uma ontologia para o DI-UM . . . . .	215
12.4	Parte dos relacionamentos do DI-UM . . . . .	217
12.5	Avaliação de Tempo de Extracção . . . . .	229
12.6	Visualização do topic map do DI no <i>Ulisses</i> . . . . .	230
13.1	Diagrama entidade-relacionamento da base de dados da <i>E-Sell</i> . . . . .	234
13.2	Especificação XTche do tipo de tópico <i>Cliente</i> . . . . .	243
13.3	Especificação XTche do tipo de tópico <i>produto</i> . . . . .	244
13.4	Especificação XTche do tipo de associação <i>ordem de compra</i> . . . . .	245
13.5	Especificação XTche do tipo de associação <i>comprar</i> . . . . .	246
13.6	Especificação XTche do tipo de associação <i>Conter</i> . . . . .	247
13.7	Especificação em XTche das restrições contextuais . . . . .	247
13.8	Especificação XTche completa para o caso de estudo de <i>E-Sell</i> . . . . .	251
14.1	Exemplo da capa de um Almanaque. . . . .	256
14.2	Um exemplo de uma história de vida retirada do mesmo Almanaque. . . . .	257
14.3	Esquema dos Almanques. . . . .	258
14.4	Excerto de um Diário de Viagem. . . . .	259
14.5	Esquema de um Diário de Viagem. . . . .	259
14.6	Diagrama Entidade-Relacionamento da base de dados dos Passaportes. . . . .	260
14.7	Ontologia para o Museu da Emigração . . . . .	265

14.8	Etapa 1 – O Projecto de Extracção. . . . .	270
14.9	Etapa 2 – Selecção do modo de armazenamento do topic map: na BD Ontologia (esquerda) ou num ficheiro XTM (direita) . . . . .	271
14.10	Etapa 3 – Dados da extracção . . . . .	271
14.11	Etapa 4 – Resultados da Extracção . . . . .	272
14.12	Restrições de esquema XTche para os tipos de tópico Emigrante, História de Vida e História de Almanaque . . . . .	272
14.13	Restrições de esquema XTche para os tipos de associação que envolvem os tópicos Emigrante, História de Vida e História de Almanaque . . . . .	274
14.14	Página principal do website do Museu da Emigração . . . . .	275
14.15	Página sobre um emigrante . . . . .	276
14.16	Visão de um Passaporte. . . . .	276
B.1	Regra XTche para a associação <i>orientar</i> referente a um <i>closed maplet pattern</i>	305
B.2	Regra XTche para a associação <i>orientar</i> referente a um <i>open maplet pattern</i>	306
B.3	Regra XTche para a verificação da existência da associação <i>nacionalidade</i> , com os papéis <i>cidadão</i> e <i>nação</i> sendo desempenhados por <i>Giovani</i> e <i>Brasil</i> , respectivamente . . . . .	307
B.4	Regra XTche para definir que todo tópico do tipo <i>pessoa</i> deve estar presente em uma associação do tipo <i>orientar</i> . . . . .	309
B.5	Regra XTche para definir que tópico do tipo <i>orientador</i> não pode ter nen- huma ocorrência . . . . .	309
B.6	Regra XTche para definir opcionalidade . . . . .	310

# Lista de Tabelas

2.1	Comparativo entre as Linguagens para <i>Semantic Web</i> . . . . .	40
7.1	Associação <i>one2one</i> entre Autor e Instituição . . . . .	125
7.2	Associações <i>MxN</i> . . . . .	127
8.1	Comparativo entre KAON REVERSE e Oveia . . . . .	148
12.1	Conceitos encontrados no domínio do Departamento de Informática. . . . .	213
12.2	Associações <i>one2one</i> . . . . .	221
12.3	Associações <i>MxN</i> . . . . .	222
12.4	Teste de desempenho do Oveia . . . . .	229
14.1	Conceitos encontrados no domínio do Museu da Emigração. . . . .	264



# Capítulo 1

## Introdução

*Querência amada  
planícies e serras  
Os braços que me puxa  
da linda mulher gaúcha  
Beleza da minha terra*

Teixeirinha

A norma ISO/IEC 13250 Topic Maps (Biezunsky, Bryan, and Newcomb, 1999) é uma norma para organização e representação de conhecimento sobre um domínio específico, que permite a descrição de temas e de seus relacionamentos. Steve Pepper (Pepper, 2000) define *tema* como um termo usado para designar alguma coisa, independentemente de ser algo concreto (por exemplo, um automóvel ou uma pessoa) ou apenas uma abstracção mental (por exemplo, um sentimento).

Um topic map<sup>1</sup> pode ser visto como um conjunto organizado de tópicos (representação formal de temas de um universo de discurso), contendo:

- uma estrutura hierárquica de tópicos (definido pelas relações *é-um* ou *contém*);
- vários nomes para cada tópico (ou tema de um índice);
- ponteiros (ocorrências) entre tópicos e documentos externos (conectando os temas aos recursos de informação);
- relacionamentos semânticos (associações) entre tópicos.

Um topic map é, basicamente, composto por tópicos e associações que dão origem a uma rede semântica estruturada que agrupa informações relacionadas sobre um certo domínio.

---

<sup>1</sup>Quando se referir a norma ISO 13250 (Biezunsky, Bryan, and Newcomb, 1999) usa-se a notação “Topic Maps”; por sua vez, quando está a referir um mapa de tópicos em particular, usa-se a notação “topic map” (ou “topic maps”, no plural).

Essa rede é vista como um grafo: as associações são os arcos, enquanto que os tópicos são os nodos.

Tal como qualquer grafo, os **Topic Maps** podem ser representados em muitas linguagens diferentes; dentre elas, podem ser escrito de acordo com uma sintaxe XML. **XML Topic Maps (XTM)** é definido por um DTD (Pepper and Moore, 2001b) que define a estrutura dos elementos que representam cada conceito da norma ISO 13250.

No âmbito de representação de conhecimento (*Knowledge Representation*), os **Topic Maps** possuem as seguintes características principais:

- Permitem estabelecer uma rede associativa estruturada entre os tópicos;
- Organizam hierarquicamente os recursos num novo domínio de informação/conhecimento, relacionando-os com os tópicos;
- Propiciam a interoperabilidade entre recursos de informação distintos, através de uma visão conceptual integrada da ontologia sobre o domínio.

De acordo com Ann Wrightson (Wrightson, 2001), **Topic Maps** é uma norma que permite a representação de ontologias. Uma ontologia é uma forma de descrever um entendimento comum partilhável, sobre o tipo de objectos e relacionamentos entre eles, de tal forma que a comunicação pode acontecer entre pessoas e sistemas. Em outras palavras, é a terminologia de um domínio, definindo o universo de discurso. As ontologias podem ser usadas para:

- Criar um vocabulário estruturado e válido, que possa ser usado por uma comunidade;
- Definir e usar relacionamentos lógicos e regras entre os temas, permitindo um uso eficiente de agentes inteligentes;
- Desenvolver, manter e publicar conhecimento (que pode sofrer mudanças rapidamente) sobre um domínio (todo ou parte dele), fornecendo facilmente visões distintas sobre o mesmo.

O desenvolvimento de uma ontologia basicamente envolve a catalogação de um conjunto de temas e relações, usando e/ou implementando um tipo de documento (para intercâmbio) e ferramentas para processar tais documentos.

A habilidade de conectar recursos, onde quer que eles estejam, e organizá-los de acordo com uma ontologia, torna a norma **Topic Maps** um componente chave para a gestão de conhecimento na Web.

Os factos acima explicam a importância de **Topic Maps** para descrever conhecimento em geral; em particular, suas aplicações para a definição de ontologias é um dos campos que está em crescimento.

Se por um lado esta introdução ajuda no entendimento de nosso interesse em **Topic Maps** na área de representação de conhecimento, por outro lado, os conceitos até aqui introduzidos



apontam para a necessidade indubitável de mecanismos que possam garantir a correcção semântica de **Topic Maps**.

Para construir sistemas confiáveis, é crucial ter certeza sobre a completa correcção da rede semântica subjacente. Assim como em outros campos, como linguagens formais e processamento de documentos, é aconselhável validar a sintaxe e a semântica de topic maps antes de seu uso. Esse é o foco principal desta tese, onde é proposta a linguagem chamada XTche, para definir esquemas e restrições em **Topic Maps**.

## 1.1 Restrições em Topic Maps

Um das principais vantagens de **Topic Maps** é a liberdade em sua criação. A informação pode ser facilmente representada num topic map, visto que a definição de um tópico é muito abrangente. Contudo, esta liberdade pode ser uma ameaça à consistência de um topic map. Consequentemente quando se está a desenvolver um topic map, os projectistas necessitam impor certas regras.

A norma **Topic Maps** não possui nenhuma forma de restringir os valores concretos e relativos que tópicos e associações podem tomar. Assim sendo, não sugere nenhum mecanismo para validar topic maps de acordo com um conjunto de regras semânticas (ou de conteúdo) definidas por utilizadores. Enquanto que a validação estrutural decorre da sua definição formal estabelecida pela norma, o processo de validar e garantir a consistência semântica de topic maps está deixada a cargo do projectista. A validação manual de um pequeno topic map pode ser realizada facilmente, porém quando trata-se de um topic map de tamanho considerável, tal processo torna-se praticamente impossível, sendo fundamental um mecanismo de validação automática.

Uma linguagem formal necessita ser definida para a especificação destas restrições sobre topic maps ser compreendida e realizada por um processo automatizado.

Um exemplo de uma validação que pode ser efectuada em topic maps, pode ser: garantir que cada associação do tipo “orientação” só contem dois tópicos participantes, onde um está desempenhando o papel de “orientador” e o outro desempenha o papel de “orientando”.

Uma linguagem chamada *Topic Map Constraint Language* (TMCL) (Nishikawa, Moore, and Bogachev, 2004) está actualmente em vias de ser normalizada pelo do comité ISO/IEC JTC1/SC34 para a definição de restrições em **Topic Maps**, sob a nomenclatura ISO 19756. O objectivo de TMCL é permitir uma especificação formal de regras para documentos topic map. No momento, TMCL não é uma linguagem concreta; pelo contrário, contém apenas indicação dos requisitos que deverão ser contemplados pelas futuras versões de linguagens concretas.

Há um considerável número de vantagens obtidas a partir da definição de uma linguagem de restrições para topic maps. Como mencionado anteriormente, uma linguagem de restrições é usada para definir regras as quais irão garantir a consistência de um topic map. Tal linguagem também auxilia a definição da semântica de topic maps.

## 1.2 A História deste doutoramento

O início dos trabalhos de doutoramento no âmbito do grupo de Especificação e Processamento de Linguagens (EPL) desta tese foi em Outubro de 2001, onde a semântica estava em foco. Naquele momento, a preocupação principal era garantir a semântica em documentos XML. Para este fim, na sequência da tese de doutoramento de José Carlos Ramalho (Ramalho, 2000) desenvolveu-se a linguagem XCSL (*XML Constraint Specification Language*) para permitir a especificação de restrições em documentos XML. Após o XML Europe 2002<sup>2</sup>, onde XCSL foi apresentada à comunidade XML e comparada com as linguagens similares – Schematron (Dodds, 2001) e XML Schema (Duckett et al., 2001) –, a pesquisa tomou um novo rumo.

Em Agosto de 2002, deu-se início à pesquisa sobre Topic Maps. Após um levantamento bibliográfico sobre o tema, partiu-se para a construção de topic maps na sintaxe XTM. O primeiro topic map – desenvolvido manualmente baseado no DTD de XTM – representava o domínio do grupo de Especificação e Processamento de Linguagens da Universidade do Minho. O topic map foi construído a partir dos dados sobre este grupo de pesquisa contidos num documento XML.

Este processo manual levou a pensar na automatização da construção. Para isso, desenvolveu-se uma folha de estilos XSL que mapeava o documento XML original para o esquema de XTM. Com isso, toda vez que se desejava inserir novos elementos no topic map, bastava acrescentar dados no ficheiro XML e processar a folha de estilos novamente.

Entretanto, para outro caso de estudo – uma biblioteca de filmes digitais – seguiu-se esta mesma abordagem de automatização e, portanto, foi necessário desenvolver uma nova folha de estilos, desta vez adequada à estrutura da fonte XML em questão. Neste passo se verificou que esta nova folha tinha características muito similares à anterior. Então decidiu-se pela definição de uma linguagem de especificação de extractores de topic maps, onde o projectista define: (1) de onde são extraídos os dados do documento XML fonte, para a montagem do topic map desejado; (2) e o domínio, na ontologia a ser representada. Essa linguagem recebeu o nome de XSTM (*XML Specification for Topic Maps*), enquanto que o extractor foi chamado TM-Builder. Foi então construído um gerador de TM-Builders a partir de especificações XSTM, o qual foi chamado de XSTM-P.

Uma vez obtido um topic map, havia a necessidade de visualizar a informação nele representado. Com essa finalidade em mente, no final de 2002 desenvolveu-se o *Ulisses* no âmbito da disciplina de *Processamento Estruturado de Documentos* dos cursos de *Licenciatura Matemática e Ciências da Computação* (LMCC) e *Licenciatura em Engenharia de Sistemas* (LESI) e *Informática* da Universidade do Minho. O *Ulisses* é responsável por produzir uma navegação Web sobre os topic maps criados e possuía, naquele momento, duas variantes: (I) geração de páginas HTML estáticas para cada tópico (através de uma folha de estilos XSL), representando as associações como *links*; (II) interpretação do topic map em tempo de navegação através de uma CGI (*Common Gateway Interface*) escrita em Perl, gerando as páginas dinamicamente de acordo com a navegação realizada pelo utilizador.

---

<sup>2</sup>XML Europe 2002 – <http://www.xml europe.com/2002/>

A partir de então, o TM-Builder foi aplicado em diversos casos de estudo, propiciando uma criação automática de topic maps a partir de fontes XML. Porém, no início de 2003 deparou-se com uma situação onde a fonte era uma base de dados relacional. Nesse caso, houve a necessidade de fazer o descarregamento dos dados para um ficheiro XML. Isso serviu de motivação para a definição de um extractor que não fosse limitado a um determinado tipo de fonte. Assim nascia o Oveia.

O Oveia tornou possível a extração de dados directamente de base de dados relacionais, pois além de uma linguagem onde se especifica a ontologia do topic map desejado (XS4TM – *XML Specification for Topic Maps*<sup>3</sup>), definiu-se uma linguagem para a especificação das fontes (XSDS – *XML Specification for Data Sources*). Deste modo, o Oveia permite que várias fontes possam ser consultadas para a criação de um único topic map, propiciando assim a interoperabilidade entre as fontes.

Este par de geradores supriu uma das carências da comunidade de Topic Maps. Contudo, percebeu-se que uma carência ainda maior derivava da ausência de um mecanismo de validação semântica de topic maps. O grau de liberdade na criação de um topic map é visto como uma das principais vantagens desta norma (permitindo a representação de qualquer domínio); contudo também pode ser visto como uma desvantagem, pois tal representação pode ser muito ampla, tornando o topic map inconsistente. Isso trouxe à tona todo o processo realizado em XCSL, porém agora voltado para o domínio da representação de conhecimento.

A fim de impedir este tipo de inconsistência, no final de 2003 decidiu-se pela definição e criação de uma linguagem para a especificação de regras que permitam validar semanticamente um topic map. Desta forma, surgia XTche. Baseada nos requisitos de TMCL, XTche tem como objectivo permitir que o projectista defina um conjunto de restrições a serem aplicadas em topic maps. Estas restrições também podem ser utilizadas para definir a estrutura de uma família de topic maps. Além disso, XTche possui uma sintaxe baseada em XML Schema, o que facilita a sua especificação num ambiente visual utilizando qualquer editor de XML Schema, como por exemplo o XML Spy®.

Tema principal da investigação durante todo o ano de 2004, a linguagem XTche foi formalizada e submetida às duas principais conferências de XML a nível mundial, sendo aceite na *XML Conference & Exhibition 2004*<sup>4</sup> e na *Extreme Markup Languages 2005*<sup>5</sup>.

Entre as principais metas traçadas para XTche destaca-se a perspectiva de uma submissão ao comité ISO/IEC JTC 1/SC34<sup>6</sup>, comité ISO responsável pelo projecto de TMCL. Essa submissão terá como objectivo contribuir com o desenvolvimento de TMCL, fornecendo uma linguagem (sintaxe e semântica) e seu processador.

De uma forma geral, os processadores desenvolvidos no âmbito deste trabalho de doutoramento – TM-Builder, Oveia, XTche e Ulisses – podem ser integrados, onde um processo

---

<sup>3</sup>As linguagens XSTM e XS4TM possuem o mesmo nome, apenas se diferenciam pelas siglas para identificar que a primeira define a criação de um TM-Builder, enquanto que a segunda especifica a extração no Oveia.

<sup>4</sup>XML Conference & Exhibition 2004 – <http://www.xmlconference.org/xmlusa/2004/>

<sup>5</sup>Extreme Markup Languages 2005 – <http://www.extrememarkup.com/extreme/2005/>

<sup>6</sup>ISO/IEC JTC 1/SC34 – <http://www.y12.doe.gov/sgml/sc34/>

toma como entrada a saída do anterior. A agregação de todos esses processadores define o ambiente **Metamorphosis**, um ambiente que permite a especificação de extracção e construção de topic maps a partir de fontes heterogêneas, a sua validação semântica e a visualização da rede semântica representada no topic map gerado.

### 1.3 A Tese e os Contributos

**Topic Maps** é uma norma para a representação do conhecimento envolvido em um determinado universo de discurso, o domínio de aplicação em causa, baseado numa rede semântica formada por tópicos e associações. A principal vantagem de **Topic Maps** é o grau de liberdade de expressão, possibilitando a descrição completa de qualquer domínio. Nesse contexto, a tese que aqui se defende é a de que é necessário um mecanismo de validação para garantir a preservação da semântica de cada topic map, com base num conjunto de restrições (estruturais e de conteúdo) definidas por um especialista no domínio.

A maior contribuição desta tese é, portanto, a definição de uma linguagem (sua sintaxe e sua semântica) para a especificação de restrições a serem aplicadas em **Topic Maps**, com o fito de estabelecer com precisão a sua conexão semântica. Além da linguagem – que respeita a proposta TMCL elaborada pela ISO – completa esse contributo o **Processador XTche**, um processador que gera **TM-Validators** a partir de especificações **XTche**. **TM-Validator** é o processador gerado responsável por validar as restrições em qualquer topic map concreto.

A linguagem utilizada neste doutoramento para a representação de topic maps é **XTM**; a linguagem de restrições proposta, **XTche**, para declarar as regras semânticas é baseada em **XML Schema**. Com isso, a composição de regras na linguagem **XTche** pode ser realizada em editores de **XML Schema**, o que facilita sua visualização.

### 1.4 Estrutura da Dissertação

A dissertação **Topic Maps: da Sintaxe à Semântica** divide-se em três partes principais. A primeira refere-se ao estado da arte, apresentando o universo de discurso onde a tese acima está inserida. A segunda parte apresenta os contributos desta tese: um ambiente que permite a definição, extracção, validação e navegação em topic maps. Em especial, o processo de validação é visto com mais profundidade, por ser considerado o maior contributo à comunidade científica. A parte final apresenta quatro diferentes casos de estudo que confirmam a utilidade da proposta aqui defendida.

A **Parte I** é composta por quatro capítulos. O Capítulo 2 insere o leitor nas áreas de trabalho onde se integram os **Topic Maps**: representação de conhecimento e **Semantic Web**. O Capítulo 3 é todo dedicado ao tema desta tese: **Topic Maps** – conceitos subjacentes, definições e representações são apresentados de acordo com a norma ISO 13250. Um relacionamento entre **Topic Maps** e ontologias é discutido no Capítulo 4.

O Capítulo 5 aborda o tema mais relevante para esta tese: a validação semântica de

**Topic Maps**, apresentando os motivos que levaram à definição de uma metodologia, e consequentemente, a uma linguagem que permita uma verificação completa do conteúdo semântico de topic maps.

A **Parte II**, composta por cinco capítulos, inicia-se com a apresentação do ambiente proposto – o **Metamorphosis** – no Capítulo 6. Este ambiente é composto pelos módulos que são apresentados nos capítulos subsequentes.

O Capítulo 7 introduz o **TM-Builder**, o qual é considerado a origem deste ambiente. Um gerador de extractores de topic maps todo baseado em XML: a fonte de dados a ser interrogada (documentos XML), o processador implementado (em XSL) e o formato do topic map gerado (em XTM). Originalmente, até se chamou ao sistema de “*stylesheet de 2º nível*” devido ao facto de existirem, na realidade, 2 níveis de processadores XSL: um que extrai os dados das fontes para criar o topic map – o **TM-Builder**; outro que gera esse extractor. O topic map é criado a partir de uma ontologia representada na linguagem **XSTM**, a qual define como a informação, que está armazenada nas fontes, deve ser mapeada para um topic map.

A fim de suprir determinadas carências do **TM-Builder**, foi desenvolvido o **Oveia**. Conforme o que será apresentado no Capítulo 8, o **Oveia** permite a extracção de topic maps em fontes heterogêneas de informação, não estando limitado apenas a documentos XML. A partir da especificação das fontes de dados em **XSDS** e de uma ontologia representada em **XS4TM**, o **Oveia** interroga directamente as fontes, extraindo os dados seleccionados e montando o topic map de acordo com a especificação **XS4TM**.

O Capítulo 9 é visto como uma sequência do Capítulo 5. Na **Parte I** será apresentado a motivação para uma linguagem de restrições em **Topic Maps**, o que será concretizado na **Parte II** com a definição da linguagem **XTche**. Portanto, o Capítulo 9 descreve o projecto de uma linguagem que possibilite a especificação formal de restrições em **Topic Maps**, permitindo expressar condições de esquema, contextuais e de existência sobre partes específicas de um topic map. A partir de uma especificação **XTche**, o seu processador (**Processador XTche**) cria um validador de topic maps (**TM-Validator**) que aplicará as regras encontradas na especificação. No final do processo, o validador é capaz de informar ao utilizador se o topic map está de acordo com as regras ou não; neste último caso, os erros são reportados para um ficheiro XML.

O Capítulo 10 tem por objectivo a visualização eficiente do conhecimento encenado encontrada em um topic map, devendo para isso permitir uma navegação livre, mas estrutural, na específica rede semântica. Isto é um assunto crítico porque topic maps podem conter milhões de tópicos e associações. Este capítulo inicialmente apresenta algumas técnicas de navegação sobre **Topic Maps**, para posteriormente descrever o **Ulisses** e suas 3 variantes para a visualização de **Topic Maps**.

Após a finalização da parte responsável por apresentar os contributos da tese, a **Parte III** fornece quatro casos de estudos que detalharão o funcionamento de cada um dos módulos do **Metamorphosis**.

A navegação propiciada pelo **Ulisses** sobre o Mapa de Conceitos associado ao programa de uma disciplina universitária é discutida no Capítulo 11. Além disso, é realizado uma

comparação da norma **Topic Maps** com os Mapas de Conceitos.

O Capítulo 12 exhibe o **TM-Builder** e o **Oveia** na extracção de um Mapa do Site do Departamento de Informática da Universidade do Minho. Seu objectivo principal é mostrar a extracção de topic maps para usá-los na representação do conhecimento que encontra-se no mapa de um site. Portanto, é a aplicação desses módulos do **Metamorphosis** na Engenharia Web, mostrando a aplicação da variante do **Ulisses** que navega sobre a BD Ontologia.

O Capítulo 13 mostra o processo de validação semântica realizado pelo **XTche** para um topic map que representa o domínio de uma aplicação de comércio electrónico. Este caso de estudo foi completamente baseado numa proposta encontrada nos requisitos de **TMCL**. Os editores da futura norma **ISO 19756** sugerem um caso onde a linguagem para restrição em **Topic Maps** deva ser aplicado, permitindo a definição de todas as regras necessárias. Portanto, o capítulo em questão mostra como se especifica, na linguagem **XTche**, cada tipo de restrição sobre um topic map que representa esse domínio.

Depois da apresentação de um caso de estudo para cada componente do **Metamorphosis**, mostrando que eles podem ser usados independentemente, o Capítulo 14 apresenta aplicação de todo o **Metamorphosis** ao caso do Museu Virtual da Emigração. O Museu da Emigração contém no seu espólio fontes de informação compostas por documentos XML e por uma base de dados relacional. A função do **Metamorphosis** é propiciar uma visão homogénea destes recursos através da criação de um topic map que represente este universo de discurso. Para isso, usa-se o **Oveia** para a extracção do topic map, a linguagem **XTche** para validá-lo e o **Ulisses** para a navegação sobre o conhecimento do domínio. O resultado é uma visão integrada do Museu da Emigração, de acordo com a ontologia especificada pelo projectista.

A conclusão recorda a tese, resume os objectivos alcançados com este trabalho de doutoramento, ressaltando a sua contribuição principal, e examina possíveis áreas para trabalhos futuros.

O Apêndice A define as regras para a validação de uma especificação **XTche** que devem ser respeitadas pelo projectista no momento da especificação. Todas as regras estão implementadas no **Validador de XTche** – processador apresentado no Capítulo 9 responsável por conferir se o código criado pelo projectista pertence à linguagem **XTche**.

O Apêndice B descreve a comparação das linguagens de especificação de restrições em **Topic Maps**: **XTche**, **AsTMa!** e **OSL**. São abordadas várias situações onde é necessária a intervenção de um mecanismo que garanta a validade semântica de um topic map e mostrado como cada linguagem proporciona este tratamento. Este apêndice é um complemento ao Capítulo 9, onde há uma secção que faz um resumo da comparação encontrada no apêndice em questão.

## Parte I

# Topic Maps: Definição e Representação





## Capítulo 2

# Representação de Conhecimento e Semantic Web

*As armas e os barões assinalados  
que da ocidental praia lusitana  
por mares nunca de antes navegados  
passaram ainda além da taprobana*

Luís Vaz de Camões

Considerando a maneira usual que os utilizadores manuseiam os navegadores da *World Wide Web* (Web), o processo baseia-se em saltar de uma página a outra baseado em palavras-chaves, ligações Web (*links*) ou em um índice.

O utilizador lê e interpreta o significado do documento encontrado – o qual foi estruturado de uma forma qualquer pelo seu criador – decidindo se tal documento é o que ele buscava; caso negativo, uma nova procura deverá ser realizada.

Todavia, o conhecimento no documento não estará necessariamente disponível ao utilizador: pode ser requerido uma base sólida sobre o assunto para entender a informação que o documento apresenta. Por exemplo, um documento sobre *desnaturação das proteínas* em biologia contém um conhecimento que a maioria das pessoas não consegue extrair. O conhecimento não pode ser capturado e extraído porque estas pessoas não conseguem interpretar as palavras contido no documento, ou seja, elas não conseguem decifrar o significado pretendido. E por quê? Porque tais pessoas não possuem um modelo conceptual suficientemente rico sobre este determinado domínio.

Por outro lado, os computadores são úteis para organização e processamento de dados, tipicamente mantendo as informações em hierarquias rígidas, enquanto a mente humana tem a habilidade especial de ligar pequenas unidades de informação de forma aleatória. Com base nesta constatação, a segunda geração da Web, cunhada como *Semantic Web* (Berners-Lee, Hendler, and Lassila, 2001), envolve a disposição dos conceitos e de seus relacionamentos dentro de um universo de discurso de forma que não apenas a sintaxe determine uma busca:

a semântica seria fundamental. Assim, um computador poderia representar associações entre coisas que poderiam parecer não relacionadas, mas que de facto, compartilham algum relacionamento.

Os motores de busca actuais, baseados em palavras-chave, requerem que o utilizador seja seu interpretador semântico, isto é, o utilizador deve compreender o conhecimento contido em um documento sem uma interpretação de sua semântica realizada por um programa.

Este capítulo discute formas de solucionar esta situação apresentando inicialmente, na secção 2.1, o que se entende no âmbito desta tese por dado, conhecimento e informação. A secção 2.2 mostra algumas metodologias para representar abstractamente o conhecimento. Uma visão geral sobre *Semantic Web* encontra-se na secção 2.3. Por fim, serão introduzidas algumas das principais linguagens para a representação de conhecimento mais utilizadas para obter-se a *Semantic Web* (secção 2.4).

## 2.1 Dado, Informação e Conhecimento

Os termos *dado*, *informação* e *conhecimento* são usualmente utilizados de formas mal-definidas, o que é aceitável em conversas coloquiais. Contudo, tal uso pode resultar em uma destruição de distinções técnicas críticas, a ponto que esses termos possam ser vistos como sinónimos.

Dados e conhecimento são simplesmente pontos finais em uma linha contínua, como sugere a Figura 2.1. No lado esquerdo, encontra-se o *dado*; do lado direito, o *conhecimento* (Park and Hunting, 2003).

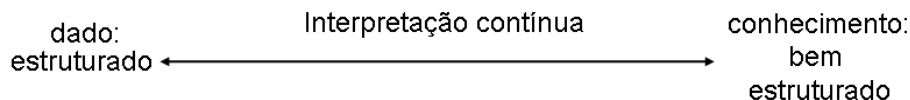


Figura 2.1: A linha contínua da interpretação. Fonte: Jack Park (2003).

O conhecimento é uma modelação simbólica complexa (uma representação) de algum aspecto de um universo de discurso (isto é, o que uma pessoa pode falar sobre seres humanos); dado é uma modelação simbólica simples. A seguinte equação expressa o relacionamento entre conhecimento e dados:

$$\text{Conhecimento} = \text{Dado} + \text{Interpretação}$$

O termo *informação* é usado nesta tese no sentido mais técnico, como Frost (Frost, 1986) derivou de Shannon (Shannon, 1974):

Information is a measure of the extent to which a piece of knowledge tells you something which you did not previously know.

Portanto, a informação contida em algum dado depende do que uma pessoa sabe sobre o tema e, em geral, isto pode variar de indivíduo para indivíduo. A informação representa mudança no conhecimento, pois ela altera o conhecimento sobre um determinado tema:

$$\text{Novo Conhecimento} = \text{Velho Conhecimento} + \text{Informação}$$

A partir disto, pode-se definir *interpretação* como o mapeamento entre um conjunto estruturado de dados e um modelo de algum conjunto de objectos em um universo de discurso com respeito ao significado pretendido para estes objectos e os relacionamentos entre os objectos.

Interpretação, por consequência, é o mapeamento entre notações: por exemplo, sequências de caracteres de algum alfabeto (para textos) ou algum conjunto de códigos binários definidos (para gráficos, vídeo, etc.), e o que tais notações pretendem significar em um universo de discurso definido para humanos. As notações são símbolos sem sentido a menos que elas tenham uma interpretação, isto é, mapeados para um objecto em um modelo. Interpretação é semântica: é interpretar as notações sintácticas de acordo com sua semântica pretendida.

Tipicamente, o modelo existe na mente de um humano. Quando os humanos *entendem*, eles simbolicamente representam os objectos de um domínio em algum modelo, assim como os relacionamentos entre esses objectos. Os humanos tem a semântica do domínio em suas mentes, a qual é bem estruturada e interpretada.

Na incapacidade de abarcar e manipular a complexidade do todo, o humano necessita de arranjar modelos (representação reduzida/parcial e normativa) para entender e trabalhar a realidade.

Quando um humano lê um documento textual, ele lê as notações nas páginas e interpreta-as de acordo com o seu modelo mental; isto é, ele fornece a semântica (o significado). Se há um desejo em difundir o conhecimento contido em um documento, é necessário torná-lo disponível a outros humanos, esperando que eles forneçam um interpretador semântico (seus modelos mentais). Contudo, não há conhecimento neste documento sem interpretação; a interpretação extrai o conhecimento a partir das notações contidas em uma página.

Se o desejo é que um computador possa auxiliar na difusão do conhecimento contido em um documento, é necessário automatizar parcialmente o processo de interpretação, o que significa que é necessário construir e representar em uma forma utilizável por computadores alguma porção de nosso modelo mental.

## 2.2 Estruturando o Conhecimento

Actualmente, diferentes soluções têm sido apresentadas para o problema de “como estruturar o conhecimento existente em um domínio”; porém, muitas têm sido rejeitadas. Há, contudo, algumas que têm prevalecido e causado uma boa impressão em todas as áreas onde o conhecimento é abundante.

Esses *sistemas de organização de conhecimento* são mecanismos que permitem uma estruturação da informação e podem ser empregados em casos onde exista uma grande colecção de dados, como museus, bibliotecas e arquivos.

Nesta secção serão apresentados as principais formas de organização de conhecimento, os quais podem ser divididos nas seguintes classes:

#### **Tipo universal para representação interna da informação :**

- Estruturas de Facetas (*Feature Structure*);

**Sistemas baseados em listas de termos:** normalmente este tipo de lista tem uma estrutura simples para a representação de conhecimento. Os exemplos mais conhecidos são:

- Dicionários;
- Índices;

**Sistemas baseados em grafos:** determinam associações entre os termos através de um conjunto de relações semânticas. Como exemplos, tem-se:

- Taxonomias;
- Thesaurus;
- Ontologias;
  - Redes Semânticas (*Semantic Network*).

Estas formas de representação do conhecimento, que são apresentadas nas subsecções seguintes, estão na origem da norma Topic Maps (Biezunsky, Bryan, and Newcomb, 1999) tal como ela hoje se apresenta.

#### **2.2.1 Estruturas de Facetas**

Em sua tese de doutoramento, José João Almeida (de Almeida, 2003) disse que a utilização de um tipo universal para representação interna da informação a manipular tem sido uma constante de quase todos os ambientes.

Para determinar qual seria esse tipo universal, vários pontos devem ser levados em consideração, como por exemplo o uso de um sistema tipado ou não (Lamport and Paulson, 1999), assim como estudar as consequências a nível do poder expressivo, da complexidade de implementação e a nível da facilidade de ligação com outros ambientes.

Com base nestes pontos, as Estruturas de Facetas (EF) – ou *Feature Structures* – apresentam-se como uma boa aproximação para a representação interna da informação. As EF foram introduzidas por Martin Kay (Kay, 1979) e rapidamente se tornaram populares como modo de representar informação em áreas de processamento de linguagem natural.

Ao descrever informação usando EF, há uma necessidade de arrumar a informação em compartimentos que associam um nome de atributo a um valor. Esse valor pode ser atômico ou uma outra EF. Considere-se o seguinte exemplo, correspondente à informação ligada à palavra *comprador* (de Almeida, 2003):

```

1  [ name -> comprador
2    género -> masculino
3    cat -> adjetivo_nomeComum
4    número -> singular
5    radical -> [
6      name -> comprar
7      Cla -> v. tr.
8      Conjugação -> 1
9      Fonética -> kôprár
10     FrasesExemplo -> <
11       0 Miguel comprou um carro novo.
12       A Filipa comprou mais um livro.
13       0 advogado comprou as testemunhas.
14     ...>
15     Derivadas -> [
16       ...
17     ...

```

A informação contida nesta FS está compartimentada em vários atributos. Certos atributos têm valores atômicos (exemplo *género*) outros têm valores complexos (exemplo *radical*).

### 2.2.2 Dicionários

Um dicionário<sup>1</sup> é uma colecção alfabética dos vocábulos de uma língua ou de qualquer ramo do saber, com a respectiva significação, carácter fonético, mórfico e sintáctico (léxico). Pode ser visto também como um mapeamento de termos para a sua descrição ou definição.

As principais características de dicionários são:

- é um documento simples para definição de termos que requerem esclarecimentos, de modo a melhorar a comunicação e a reduzir o risco de mal-entendidos;
- é criado durante o levantamento de requisitos, e é continuamente aperfeiçoado em cada ciclo de desenvolvimento, a medida que novos termos são encontrados;
- é feito em paralelo às especificações de requisitos, aos casos de uso e ao modelo conceptual.

Um dicionário pode ser visto como um vocabulário explicativo dos termos, conceitos, palavras, expressões, frases utilizadas no domínio de aplicação em questão e que podem dar margens a interpretações erradas ou que sejam desconhecidas do público alvo e não tenham sido explicados no texto.

<sup>1</sup>Para uma definição do termo *dictionary*, veja *The American Heritage Dictionary of the English Language: Fourth Edition*, em <http://www.bartleby.com/61/88/D0208800.html> – Acedido em 15/11/2004

Um tipo de dicionário é o glossário. Glossário<sup>2</sup> é um vocabulário em que se dá a explicação de certas palavras antigas ou pouco conhecidas sobre um determinado tema. Também pode ser visto como um dicionário de termos técnicos de uma arte ou ciência.

Em vez de um apontador para uma ocorrência de um conceito, como faz um índice, um dicionário apenas fornece a definição do conceito. Ele pode conter informações adicionais, tais como as referências *veja (see)* e *veja também (see also)*, ou fornecer um guia sobre o uso ou pronúncia da linguagem em questão.

### 2.2.3 Índices

Um índice<sup>3</sup> é uma lista em ordem alfabética dos termos relevantes que surgem em um documento, ou conjunto de documentos, mapeados para a sua localização nos referidos documentos, ou seja, o índice mapeia os termos para os locais onde aparecem a informação sobre eles.

Por exemplo, em um livro indica-se a página (ou páginas) onde cada termo é mencionado; numa biblioteca, indica-se a posição (sala, estante, prateleira) de cada obra descrita pelo seu título, autor, ou assunto. Já numa base de dados, associa-se a chave de cada registo com a sua localização física no disco. No caso de um índice remissivo em um livro, cada conceito é associado com a lista de páginas onde tal conceito foi definido.

Ao contrário dos 3 modelos de representação de conhecimento que serão apresentados a seguir – os quais descrevem relações entre termos – o índice mapeia termos em posições (o que também facilita a busca).

### 2.2.4 Taxonomias

Taxonomia (do Grego: *taxis* significa organização ou divisão e *nomos* significa lei) é a ciência de classificação.

Uma boa taxonomia<sup>4</sup> apresenta somente uma dimensão. Em taxonomia, as categorias devem ser mutuamente exclusivas, assim como um conceito individual deve ser encontrado em um local somente. A taxonomia deve ser exaustiva; todas as possibilidades devem ser incluídas.

Uma das mais conhecidas taxonomias é a inventada por um cientista sueco, Carolus Linnaeus (1707-1778), na qual ele classifica os organismos vivos (Prescott, Harley, and Harley,

---

<sup>2</sup>Para uma definição do termo *glossary*, veja *The American Heritage Dictionary of the English Language: Fourth Edition*, em <http://www.bartleby.com/61/66/G0156600.html> – Acedido em 15/11/2004

<sup>3</sup>Para uma definição do termo *index*, veja *The American Heritage Dictionary of the English Language: Fourth Edition*, em <http://www.bartleby.com/61/7/I0100700.html> – Acedido em 15/11/2004.

<sup>4</sup>Para uma definição do termo *taxonomy*, veja *The American Heritage Dictionary of the English Language: Fourth Edition*, em <http://www.bartleby.com/61/54/T0065400.html> – Acedido em 15/11/2004. é um sistema de classificação, organizando entidades/conceitos. Enquanto a ontologia descreve completamente o domínio e os diferentes conceitos que compõem este domínio, uma taxonomia descreve o relacionamento hierárquico entre os conceitos, identificando os membros das classes e subclasses

1996). Considera-se que o seu trabalho está na origem da botânica moderna e da nomenclatura zoológica.

Uma classificação divide os conceitos que compõem um domínio em diferentes classes, assim como define os relacionamentos entre estes diferentes conceitos. Esses relacionamentos dizem muito sobre o significado dos conceitos individuais.

Um exemplo de uma taxonomia é apresentado na Figura 2.2. Esta taxonomia representa uma hierarquia dos conceitos encontrados no domínio do planeta Terra. As relações estão definidas através de ligações entre os conceitos: os conceitos inferiores são sub-classes dos conceitos acima deles. As sub-classes herdam as características de suas classes. Assim sendo, a partir da taxonomia da Figura 2.2 pode-se dizer que um humano é um mamífero, que por sua vez, é um animal<sup>5</sup>.

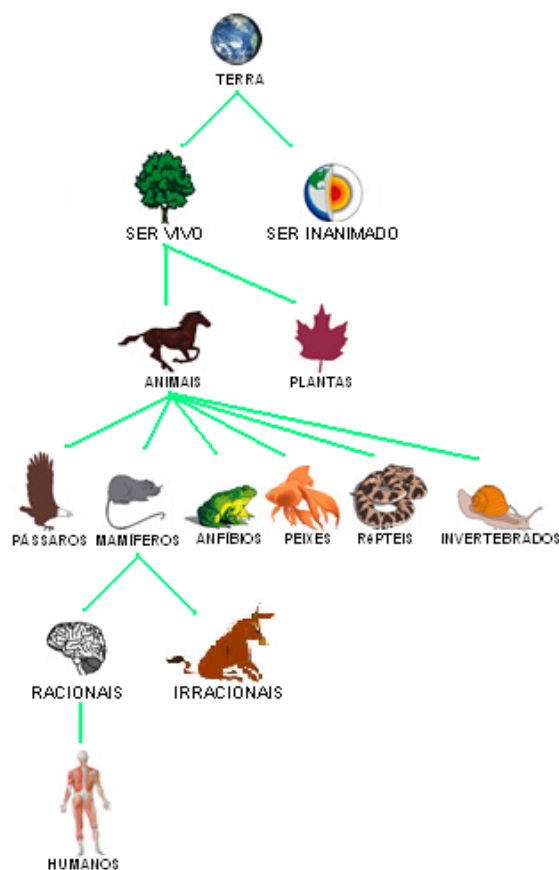


Figura 2.2: Uma taxonomia para o planeta Terra.

Em uma taxonomia, as sub-classes herdam propriedades das classes acima delas (super-classes). Por exemplo, qualquer animal pode ser considerado um ser vivo; mas não pode, ao mesmo tempo, ser planta ou animal; isto significa que ambos partilham as propriedades

<sup>5</sup>Figura adaptada a partir da original encontrada em <http://www.labschool.org/hondasite/taxonomy.html>

dos organismos, mas depois cada um tem suas características distintas. Todos os filhos de um nodo pai são disjuntos. Todos os nodos filhos herdam informações dos nodos pai. Um nodo filho pode ter mais propriedades que o seu nodo pai; e, no mínimo, ele terá todas as propriedades de seu nodo pai.

### 2.2.5 Thesauri

Thesauri<sup>6</sup> basicamente estende as taxonomias a fim de torná-las mais completas para a descrição de um domínio, permitindo outros tipos de relações entre as classes, além de uma simples hierarquia (Garshol, 2004a).

Além da sua capacidade de organização, um thesaurus também tem um valor didático, porque utiliza conceitos específicos da área de conhecimento que contempla e permite, por meio das relações entre os termos, uma melhor compreensão da área.

Um thesaurus pode então ser visto como uma rede de termos interrelacionados através de um conjunto (geralmente não muito grande) de relações semânticas dentro de um domínio particular. A rede contém referências cruzadas e fornece as associações entre eles. Dado um certo termo, o thesaurus indica quais os termos que têm o mesmo significado, qual a super-classe, as sub-classes, etc.

Em suma e de acordo com a norma ISO 2788, 1986:2 standard (1986) (ISO, 1986):

Thesaurus is the vocabulary of a controlled indexing language, formally organized so that the a priori relationships between concepts (for example as *broader* and *narrower*) are made explicit.

Um *thesaurus* é um índice controlado utilizado tanto na indexação, quanto na recuperação de documentos. Situando-se entre a ontologia (qualquer relação entre termos) e a taxonomia (relação hierárquica), o thesaurus descreve relações de sinonímia e hierarquias, das quais se destacam:

**Uso (*Use*):** refere outro termo que deve ser seleccionado ao invés de um certo termo. Significa que ambos os termos são sinónimos;

**Termo Genérico ou Superior:** relação hierárquica que indica a super-classe (termo genérico);

**Termo Específico:** relação hierárquica que indica as sub-classes (termos específicos);

**Termo Relacionado:** refere outro termo relacionado de forma não hierárquica que está relacionado com um certo termo, sem ser um sinónimo (*Uso*);

**Nota de Contexto (*Scope Note*):** um texto anexado ao termo explicando seu significado dentro do thesaurus.

---

<sup>6</sup>Para uma definição do termo *thesaurus*, veja *The American Heritage Dictionary of the English Language: Fourth Edition*, em <http://www.bartleby.com/61/6/T0160600.html> – Acedido em 15/11/2004.



Um exemplo de um thesaurus está descrito na Figura 2.3. Este thesaurus é baseado na taxonomia da Figura 2.2, referindo-se também aos conceitos encontrados no domínio do planeta Terra. Além das relações definidas na taxonomia (do tipo “é-um”), este thesaurus contém relações do tipo “termo relacionado” (ligando termos com algo em comum) e “uso” (apresentando um uso de uma classe).

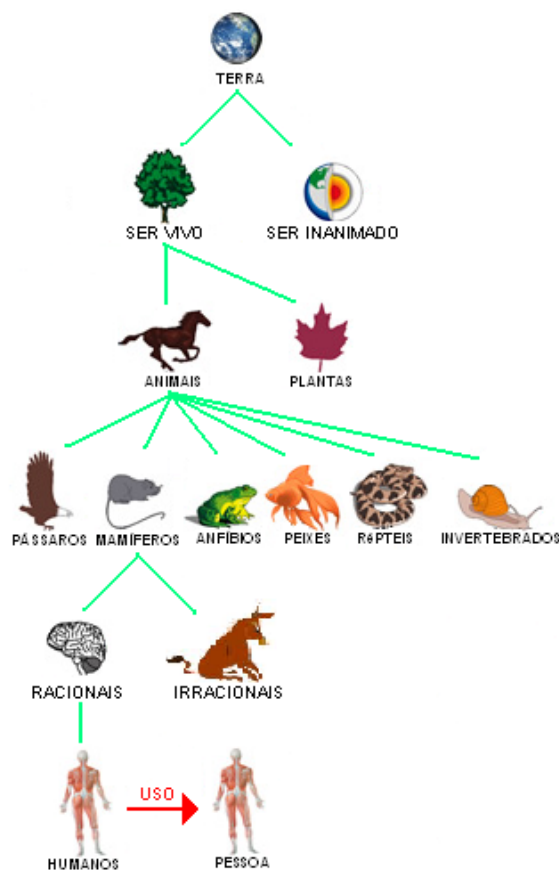


Figura 2.3: Um thesaurus para o planeta Terra.

Um bom exemplo de thesaurus é o da UNESCO (UNESCO, 1995). O Thesaurus UNESCO é um vocabulário controlado<sup>7</sup> desenvolvido pela *University of London Computer Centre* (ULCC) com permissão da *United Nations Educational, Scientific and Cultural Organization* (UNESCO), que inclui a indexação e pesquisa nas diversas áreas do conhecimento contidas nas bases de dados da UNESCO.

<sup>7</sup>Vocabulário controlado é um documento que especifica quais são os termos preferidos para serem usados domínio em questão. Padronizando os termos, é possível evitar problemas de inconsistência nas buscas realizadas. Por exemplo, quando se procura pela palavra-chave “professores de história”, podem não aparecer resultados relevantes porque o documento que continha a informação desejada possui o termo “docentes” ao invés de “professores”.

### 2.2.6 Ontologias

Hoje em dia, os sistemas convencionais de consulta utilizam técnicas de base sintáctica sobre uma forma de concordância léxica, mais do que uma aplicação da base de conhecimento do campo de interesse. Em muitos domínios, o utilizador está interessado em encontrar informação onde a relevância dos documentos não pode ser medida através do uso de sistemas de busca por palavras chaves. Neste contexto, algumas propostas envolvem a criação de metadados que envolvem a noção de ontologia.

Dentro de uma mesma área podem ser encontradas diferentes definições e caracterizações de **ontologia**<sup>8</sup>. Na área de Inteligência Artificial, Guarino define a ontologia como uma caracterização axiomática do significado do vocabulário lógico (Guarino, 1997); para Sowa, a ontologia define os tipos de coisas que existem no domínio de uma aplicação (Sowa, 2000).

Segundo Gruber (Gruber, 1993), uma ontologia é uma especificação explícita de uma conceptualização. Também pode ser entendida como um conjunto de termos hierarquicamente estruturados para a descrição de um domínio, o qual pode ser utilizado como um esqueleto fundamental para uma base de conhecimento.

Na área de Sistemas de Informação, na qual se encaixa esta dissertação, ontologia é definida como um conjunto de conceitos e termos ligados entre si (numa rede) que podem ser usados para descrever alguma área do conhecimento ou construir uma representação para o conhecimento (Swartout and Tate, 1999). Segundo Chandrasekaran (Chandrasekaran, 1999), ontologias são teorias de conteúdo sobre os tipos de objectos, propriedades de objectos e relacionamentos entre objectos que são possíveis em um domínio de conhecimento específico.

Uma ontologia também pode ser vista como uma teoria lógica para descrever o significado pretendido de um vocabulário formal, isto é, seu comprometimento com uma conceptualização particular de um domínio. Estas incluem estruturas que permitem manipular termos de uma forma muito eficiente e útil para o utilizador e mecanismos de validação para comunicação inter-programas. A importância de seu uso é devida à capacidade de representar hierarquias de classes de objectos (taxonomias) e seus relacionamentos.

As ontologias colaboram no sentido de se obter uma Web onde os recursos disponíveis são acessíveis não somente por seres humanos, mas também por processos automatizados. Elas podem ser vistas como metadados que representam explicitamente a semântica dos dados, de forma processável por máquina, tornando-se factor chave para o desenvolvimento da *Semantic Web*. Isto reflecte a visão de Berners Lee (Berners-Lee, Hendler, and Lassila, 2001).

Os sistemas de raciocínio baseados em ontologias podem operacionalizar esta semântica no sentido de agregar vários serviços à Web. As ontologias ajudam as pessoas e os computadores tanto a aceder a informação que eles precisam quanto a comunicarem-se entre si de forma efectiva. Isto é possível tornando explícita a ligação entre a forma e o conteúdo da informação. As ontologias têm, portanto, um papel crucial no sentido que permitem

---

<sup>8</sup>Para uma definição do termo *ontology*, veja *The American Heritage Dictionary of the English Language: Fourth Edition*, em <http://www.bartleby.com/61/32/O0083200.html> – Acedido em 15/11/2004.

o acesso, a interoperabilidade e a comunicação baseados em conteúdo, fornecendo à Web um nível de serviço qualitativamente novo.

O desenvolvimento de ontologias irá fornecer o mecanismo de construção da parte semântica da *Semantic Web*. O modelo em camadas proposto por Berners-Lee (Berners-Lee, 2000) tem sido aceite principalmente como representação para a arquitectura da *Semantic Web*. O desenvolvimento de tais mecanismos depende, obrigatoriamente, de linguagens que expressem a informação de maneira a ser entendida por máquinas. O desafio é proporcionar uma linguagem que manipule igualmente, de maneira eficiente, dados e regras para deduções sobre esses dados e que permita que regras existentes em qualquer sistema de representação de conhecimento possam ser exportadas para a Web.

O desenvolvimento de ontologias deverá representar uma parcela significativa de esforço no desenvolvimento de qualquer aplicação no futuro. Dessa forma, o desenvolvimento de ambientes para construção e manipulação de ontologias é fundamental. Tais ambientes devem ser compostos de um repositório de ontologias que possa ser manipulado por projectistas, utilizadores e programas de aplicação, permitindo a navegação, pesquisa e reutilização de termos. Quando novos termos forem acrescentados à ontologia, o ambiente deve verificar a consistência do repositório.

Uma ontologia define os termos usados para descrever e representar uma área de conhecimento. A ontologia formaliza o conhecimento através da utilização de cinco componentes (Noy and McGuinness, 2001):

1. Conceitos que são a representação de algo, ou de qualquer coisa, acerca do domínio em questão. As propriedades de um conceito são denominadas de atributos. Por exemplo, um conceito poderia ser uma pessoa, tendo como atributo a idade;
2. Relacionamentos que são as interligações entre os conceitos do domínio. Por exemplo, pode-se ter um relacionamento entre pessoa e universidade, através da relação “estuda em”;
3. Propriedades das classes, e seus valores permitidos;
4. Axiomas que representam as condições que irão restringir a interpretação dos conceitos e relações. Eles modelam predicados que terão de ser sempre verdadeiros. Por exemplo, define-se que a idade de uma pessoa corresponde à data actual subtraindo a data de seu nascimento;
5. Instâncias que representam os elementos de uma ontologia, ou seja, são as concretizações dos conceitos e relações que foram estabelecidas pela ontologia.

Assim, pode-se dizer que a ontologia visa (em alguns aspectos) desenvolver um conjunto de regras que possibilitem a interpretação das informações disponibilizadas na Web, ou seja, a extracção do seu significado. Resumidamente, a utilização de ontologias oferece vantagens como: possibilitar o compartilhamento e a interoperabilidade do conhecimento entre diferentes domínios (mais detalhes no capítulo 8); estruturar o domínio de forma

que se permita sua compreensão com maior clareza e objectividade; reutilizar conceitos em diferentes domínios.

Baseando-se uma vez mais no planeta Terra, uma ontologia para este domínio apresenta-se na Figura 2.4. Além das relações definidas no thesaurus descrito na Figura 2.3, a ontologia permite a inclusão de todo e qualquer relacionamento que possa ser encontrado no domínio em questão, além de permitir a inclusão de propriedades às classes. Neste exemplo, é efectuada uma relação denominada “come” entre “humano” e “peixes”, definindo que “humano come peixe”.

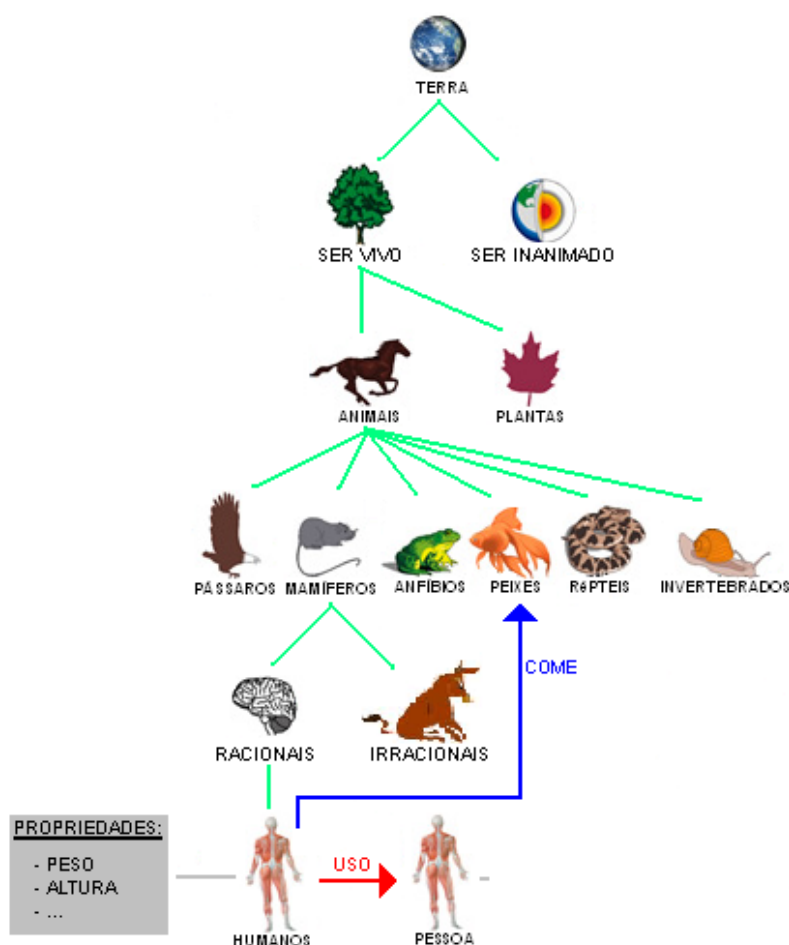


Figura 2.4: Uma ontologia para o planeta Terra.

Um exemplo mais complexo é o de uma ontologia sobre um domínio relativo ao *conserto de computadores*, a qual pode incluir certos conceitos, como disco rígido (*hard disk*), unidade de disquete, placa-mãe, CPU e unidade de CD-ROM. Esta ontologia pode conter certas declarações como: “o técnico repara o computador” e “o computador é reparado pelo técnico”. Então, os termos *técnico*, *repara*, *é reparado por* e *computador* necessitam estar contidos também na ontologia, desempenhando papéis quer de conceito, quer de relação. Assim, o termo *repara* introduz um relacionamento entre os conceitos *técnico* e

*computador*, tal e qual como o termo *é reparado por*.

### Linguagens tradicionais para a representação de Ontologias

De acordo com Guarino (Guarino, 1997) e Gruber (Gruber, 1995), uma ontologia representa um vocabulário comum de um domínio. Assim, define o significado dos termos e as relações entre eles, organizados em uma taxonomia e contém primitivas de modelação como classes, relações, funções, axiomas e instâncias.

As principais linguagens para a representação de ontologias são:

**Lógica de Predicados:** Cycl (Lenat and Guha, 1990);

**Frame :**

- Ontolingua (Farquhar, Fikes, and Rice, 1996);
- F-Logic (Kifer, Lausen, and Wu, 1995);
- CML (*Conceptual Modeling Language*) (Schreiber et al., 1994);
- OCML (*Operational Conceptual Modeling Language*) (Motta, 1999);

**Lógica descritiva:** Loom (McGregor, 1991);

**Outras:** Telos (Mylopoulos et al., 1990).

Porém, esta dissertação aborda ontologias para Web, onde as normas principais são: RDF (*Resource Description Framework*), RDF Schema, OIL (*Ontology Inference Layer*), DAML+OIL (*DARPA Agent Markup Language*), XOL (*Ontology Exchange Language*), SHOE (*Simple HTML Ontology Extensions*), *Topic Maps* e OWL (*Ontology Web Language*). Tais linguagens serão detalhadas na Secção 2.4.

É importante ressaltar que existem diferentes conexões entre os componentes da ontologia, seus paradigmas de representação do conhecimento e suas linguagens de representação.

### Rede Semânticas (*Semantic Networks*)

Em *Redes Semânticas*, tão faladas na área de Inteligência Artificial (IA), os objectos são representados como nodos de um grafo, com relações entre objectos sendo representadas por arcos. Os nodos são organizados em uma estrutura taxonómica e os arcos representam relações binárias. Tudo que pode ser expresso em lógica de primeira ordem (van Dalen, 1994; Hodges, 1997) pode também ser expresso como uma rede semântica. Um exemplo de uma rede semântica é a WordNet<sup>9</sup> (Fellbaum, 1999).

Com as Redes Semânticas, pretende-se em IA descrever uma estrutura para representar conhecimento através da organização de conceitos. Esta subsecção tem a intenção de afirmar que as Redes Semânticas são um caso particular de ontologias.

<sup>9</sup>Para mais informações sobre WordNet, veja <http://www.cogsci.princeton.edu/~wn/>

### 2.2.7 Conexão entre os modelos

Ontologias, thesauri, taxonomias, índices e dicionários são similares nos seguintes aspectos:

- São abordagens para estruturar, classificar, modelar e representar conceitos e relacionamentos pertencentes a algum tema de interesse significativa para alguma comunidade;
- Permitem que uma comunidade adote e use o mesmo conjunto de termos de um modo uniforme;
- O significado dos termos são especificados de alguma maneira a um certo nível;

Portanto, há uma forte conexão entre os modos de expressar conhecimento descritos nas secções anteriores, conforme se pretende esquematizar na Figura 2.5. Os conceitos e seus relacionamentos são descritos e definidos em modos diferentes em ontologias, thesauri e taxonomia, como mostra a Figura 2.5. Diferentes aspectos de estrutura de conhecimento são implementados em cada um deles.

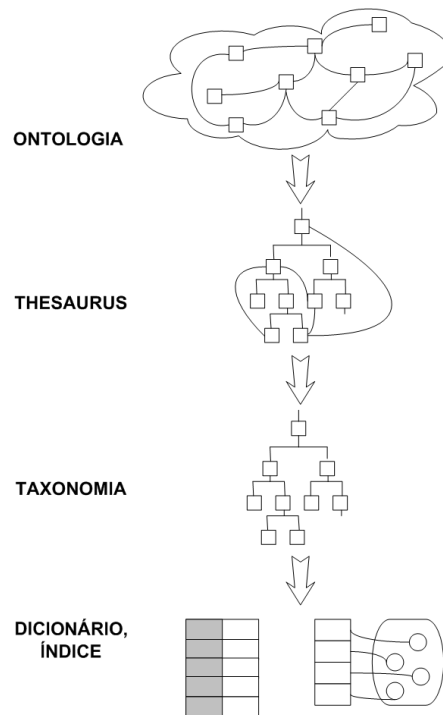


Figura 2.5: Relacionamento entre ontologia, thesaurus, taxonomia, índice e dicionário

Uma ontologia descreve os conceitos existentes em um mundo particular; em uma ontologia, todos os conceitos relevantes para um certo domínio são definidos à custa de qualquer relação binária que se julga interessante citar, enquanto que nos thesauri só se usam certas relações específicas. A taxonomia clarifica os relacionamentos hierárquicos entre os conceitos, criando uma estrutura de classes/subclasses. Os índices fornecem os apontamentos

para os locais onde cada conceito é referenciado, enquanto que os dicionários fornecem as definições dos conceitos.

## 2.3 Semantic Web

A *World Wide Web* (Web) é sem dúvida um dos maiores sucessos na história dos empreendimentos humanos, contando com utilizadores de todo o mundo, manipulando e acedendo uma enorme quantidade de informação. A Web está em amplo crescimento não apenas em seu tamanho; a sua complexidade cresce na mesma proporção. Porém, este crescimento não se reflecte no estado actual das tecnologias utilizadas para sua manipulação. A maior parte das tarefas de interpretação, acesso, extracção e manutenção da informação ainda é deixada a cargo dos utilizadores.

Os motores de busca são ineficientes quando se trata de fazer inferências complexas e relacionar assuntos aparentemente disjuntos. A simples anotação de páginas HTML por intermédio das tags <META>, ou mesmo o emprego de padrões de metadados<sup>10</sup>, não é suficiente para incluir a semântica desejada, que possibilitaria a execução de tarefas mais sofisticadas e mais úteis do que as actualmente existentes.

Na abordagem de Tim Berners-Lee (Berners-Lee, Hendler, and Lassila, 2001), esse tipo de construção leva a limitações e a um tratamento trivial do conteúdo das páginas Web por parte dos actuais browsers – limita-se a um cabeçalho; mas, em geral, as ferramentas não processam o conteúdo semântico das informações contidas em uma página.

Suponha-se que se queria pesquisar na Web a seguinte expressão: “*em quais cidades de Portugal está a chover neste exacto momento*”. Mesmo havendo hoje uma infinidade de motores de buscas e esses fazendo uso de algoritmos cada vez mais inteligentes, provavelmente uma consulta complexa do tipo “*cidades Portugal chuva agora*” não retornaria nenhum resultado. Se, de outra forma, esta consulta for feita com as palavras “*cidades Portugal chuva*”, seriam listados milhares de websites<sup>11</sup>; isto se dá porque os motores de busca actuais se preocupam apenas com a existência das referidas palavras-chave na página pesquisada e não com a semântica de tal busca. A menos que alguém possa aceder os sites encontrados, um a um, fazendo ligações entre os resultados, não se teria ainda resposta para essa pesquisa.

A diferença entre as duas frases acima descritas é o termo *agora*. Para encontrar o conjunto de sites que possuem a resposta para a primeira frase, o *agora* foi incluído pensando em sua semântica: o importante é saber em que cidades de Portugal está a chover *neste exacto instante*, e não onde choveu em algum momento no passado. Portanto, essa procura deveria ser baseada na semântica de cada uma das expressões utilizadas, não apenas na sintaxe, como é o caso dos motores de busca actuais.

---

<sup>10</sup>Metadados são dados que descrevem dados mais complexos. Um catálogo de biblioteca é um bom exemplo de metadados porque nos permite obter dados sobre os livros contidos na biblioteca. Os metadados dão informações sobre quem escreveu, quando foi publicado, que assunto é discutido, etc.

<sup>11</sup>No dia 29 de Março de 2005, foram encontrados 64.100 websites no Google.com a partir da busca pelas palavras-chave “*cidades Portugal chuva*”.

Com base nessas premissas, surgiu a ideia da *Semantic Web*, na qual o conhecimento da Web é armazenado por meio da utilização de (meta) dados processáveis por ferramentas. Pretende-se que a *Semantic Web* não seja separada da Web, mas uma extensão desta tecnologia.

Tim Berners-Lee vê a necessidade de evolução da Web, até que ela tenha o poder de fazer com que as informações possuam formato tal que as ferramentas venham a fazer associações entre informações que se relacionam (Berners-Lee, Hendler, and Lassila, 2001). Quando isso ocorrer de facto, terá sido implementada a *Semantic Web*.

Basicamente, os mecanismos a serem desenvolvidos para o estabelecimento da *Semantic Web* compreendem duas vertentes:

1. a disponibilização de uma colecção de dados estruturados e regras de inferência associadas a essa colecção;
2. a criação de ferramentas capazes de percorrer a Web realizando tarefas complexas com base nessas estruturas de conhecimento.

Para se conseguir uma *Semantic Web*, é necessário desenvolver mecanismos para por conhecimento (semântica<sup>12</sup>) na Web. Se for aceite que o conhecimento humano é baseado no entendimento de significados compartilhados, então isto segue o que a *Semantic Web* necessita envolver um mecanismo pelo qual estes significados compartilhados possam ser representados e aplicados.

### 2.3.1 Arquitectura da Semantic Web

Considere a estrutura de *Semantic Web* apresentada na Figura 2.6 (Berners-Lee, 2000). Percebe-se que Unicode<sup>13</sup> e URI (*Uniform Resource Identifier*)<sup>14</sup> constituem a base para a legibilidade e o endereçamento na *Semantic Web*.

Acima disto, há o XML e os **namespaces**: é sabido que os browsers actuais suportam XML; para os antigos, o XML pode ser transformado em HTML através de folhas de estilo XSL. Esta camada é vista como a *camada sintáctica*.

Acima destes dois níveis, entra-se no contexto da representação dos dados e dos metadados e o seu esquema. RDF, juntamente com RDF Schema (RDFS), fornece uma estrutura bem adaptada para esta necessidade. A sugestão que tem sido feita é que a especificação Topic Maps também pode satisfazer esta necessidade.

De facto, há discussões para encontrar uma associação entre RDF e Topic Maps (como apresentado no slide 15 de (Berners-Lee, 2000)); para já, as diferenças entre estas duas

<sup>12</sup>Para uma definição do termo *semantics*, veja *The American Heritage Dictionary of the English Language: Fourth Edition*, em <http://www.bartleby.com/61/83/S0248300.html> – Acedido em 15/11/2004.

<sup>13</sup>O Unicode fornece um único número para cada carácter, não importa a plataforma, não importa o programa, não importa a língua. <http://www.unicode.org/> – Acedido em 31/03/2005.

<sup>14</sup>Uma URI fornece um simples e extensível método para identificar um recurso. <http://www.gbiv.com/protocols/uri/rfc/rfc3986.html> – Acedido em 31/05/2005.



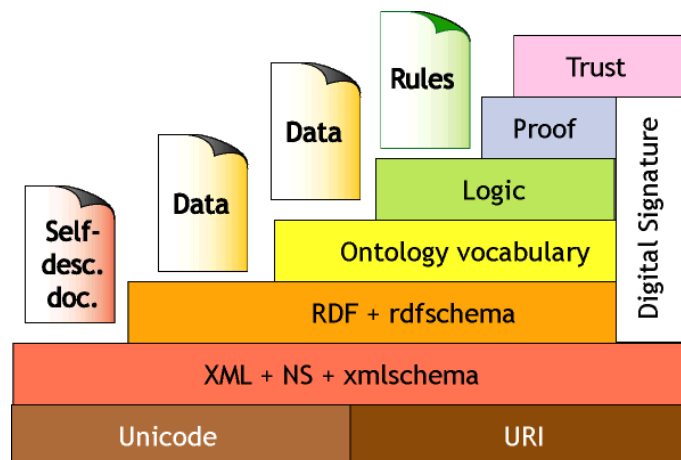


Figura 2.6: Arquitetura da *Semantic Web*. Fonte: Tim Berners-Lee (2000).

normas são debatidas na Secção 3.5. Topic Maps propicia a criação de *Published Subject Indicators* (PSIs)<sup>15</sup>, algo que pode ser obrigatório – para estabelecer uma semântica precisa da sintaxe XTM – ou algo que pode ser opcional – servindo como um tipo de ontologia para os utilizadores de Topic Maps.

Essa tese é centrada na noção de que Topic Maps oferece tais mecanismos para satisfazer, pelo menos, parte destas necessidades. Portanto, deve-se considerar que além de RDF, há outras actividades em desenvolvimento para atingir as mesmas necessidades.

No nível da camada ontológica, encontram-se linguagens que permitam a especificação de ontologias. *Ontology Inference Layer* (OIL) (Fensel et al., 2001) é uma linguagem de inferência e representação baseada na Web, que combina a utilização de modelação de primitivas provenientes das linguagens baseadas em frames com a semântica formal e, ainda com serviços de raciocinador provenientes de lógicas de descrição. OIL é um exemplo particular de aplicação de RDFS, iniciado como um projecto Europeu para trazer informação semântica para a Web. *DARPA Agent Markup Language* (DAML) (DARPA, 2001) é um projecto americano que foi integrado com OIL, resultando em DAML+OIL (Connolly et al., 2001), o qual define uma série de construções específicas para representação de ontologias em RDF.

O W3C está actualmente envolvido no desenvolvimento de um padrão para representação de ontologias, o *Web Ontology Language* (OWL) (Bechhofer et al., 2002) – que é amplamente baseado no DAML+OIL. OWL adiciona mais vocabulário para a descrição das classes e propriedades como, por exemplo, relações entre estas classes, cardinalidades, igualdades, tipos e características mais apurados das propriedades e enumerações das classes.

Finalizando a visão de *Semantic Web* da W3C apresentada na Figura 2.6, há ainda as camadas superiores, as quais ainda estão sob desenvolvimento: a camada lógica (*Logic*) expressa conhecimento através de regras, enquanto a camada de prova (*Proof*) utiliza

<sup>15</sup>Veja em <http://www.topicmaps.org/xtm/1.0/core.html>.

essas regras para inferir conhecimento. A camada confiança (*Trust*) fornece mecanismos para determinar o grau de confiança do conhecimento obtido. As assinaturas digitais (*Digital Signature*) introduzem várias camadas para garantir segurança, através do uso de codificação dos dados (*encryption*) e assinaturas digitais.

## 2.4 Linguagens para a Semantic Web

Após ter apresentado os modelos abstractos para descrever o conhecimento (Secção 2.2), é fundamental referir linguagens concretas para se poder usar tais modelos; ou seja, propor sistemas de notação para fazer descrições que os humanos e as máquinas entendam. Para isso, apresenta-se nesta secção algumas propostas no âmbito do processamento de documentos que são normas internacionais, nomeadamente RDF, RDF Schema, Topic Maps, XOL, SHOE e OWL.

### 2.4.1 Resource Description Framework – RDF

Basicamente, RDF – *Resource Description Framework* (Lassila and Swick, 1999) – define um modelo de dados para uma descrição semântica processável por computador, o qual se parece com um diagrama de entidades e relacionamentos, em que as entidades são recursos e as propriedades RDF são representadas por relacionamentos entre recursos.

Uma declaração RDF pode ser representada através de grafo pesado (etiquetado) orientado. A Figura 2.7 ilustra esta representação diagramática.



Figura 2.7: Esquema de representação da RDF em grafo

Os arcos representam as propriedades, enquanto que os nós representam os recursos. Quando um nó é um literal, é representado por um rectângulo. A direcção da seta é relevante: o arco sempre começa no sujeito e aponta para o objecto da declaração.

### O modelo básico da RDF

Uma descrição RDF assenta em três tipos de objectos, que constituem a base deste modelo:

**Recursos:** um recurso é qualquer dado ou fonte de informação que se quer descrever em RDF. Pode ser uma coisa (física ou lógica), uma tabela, uma base de dados, uma página da Web, um website inteiro ou parte deste. Pode ser também um objecto não acessível de maneira electrónica, como um livro, uma revista ou um CD. Recursos são sempre especificados por URI's.

**Propriedades:** uma propriedade é uma característica (atributo) ou uma relação utilizada para descrever o recurso.

**Declarações:** uma declaração é um recurso mais as propriedades desse recurso, mais o valor de cada uma dessas propriedades; o valor é outro recurso.

Essas três partes individuais são chamadas respectivamente de **sujeito**, **predicado** e **objecto**. Em outras palavras, o modelo básico primitivo RDF consiste em triplas formadas por: objecto, propriedade e valor.

Como se disse acima, cada tripla pode ser representada por um grafo, com o objecto e o valor sendo nodos e as propriedades etiquetas do ramo orientado do objecto para o valor. Isto pode também ser representado através de grafos rotulados como na figura 2.8.



Figura 2.8: Diagrama de nós e arcos de RDF

## Exemplos

Considere a seguinte frase: *Real Madrid possui o atleta Ronaldo*. Esta frase tem as seguintes partes:

<b>Sujeito (recurso)</b>	Real Madrid
<b>Predicado (propriedade)</b>	possui atleta
<b>Objecto (literal)</b>	Ronaldo

Este diagrama pode ser lido, de maneira mais geral, como: “<sujeito> <predicado> <objecto>”.

Nesse exemplo, considere que se quer acrescentar mais características ao objecto da declaração (Ronaldo). Observe a seguinte frase: *A pessoa, cujo nome é Ronaldo, possui o e-mail r9ronaldo@realmadrid.com e é um atleta*.

A diferença entre a frase anterior e essa é que na primeira temos uma declaração com uma única propriedade ligada a um objecto (um literal). Já nessa segunda frase, há uma propriedade estruturada (atleta) que possui duas outras propriedades (nome e e-mail).

Em RDF, propriedades estruturadas são representadas como outro recurso. Assim, de acordo com as frases acima, os dois recursos – *Real Madrid* e *Ronaldo*<sup>16</sup> – estão relacionados pelo predicado *atleta*. Textualmente, pode-se representar este domínio da seguinte forma:

O *Real Madrid* possui um *atleta* que tem o nome *Ronaldo* e o e-mail **r9ronaldo@realmadrid.com**.

<sup>16</sup>A URI que serve como chave única para o Real Madrid é <http://www.realmadrid.com>, enquanto que para o atleta *Ronaldo* é <http://www.r9ronaldo.com/>.

O grafo RDF que representa este domínio é apresentado na Figura 2.9.

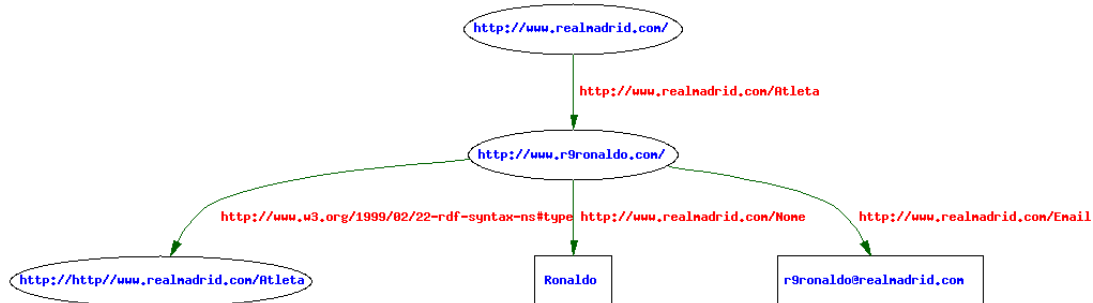


Figura 2.9: Grafo RDF exemplo

### Sintaxe de RDF/XML

O World Wide Web Consortium (W3C) (W3C, 2005) propõe uma especificação XML para codificação de um modelo de dados em RDF: a sintaxe de serialização RDF/XML (Beckett and McBride, 2004).

Para exemplificar esta sintaxe, será usada a seguinte frase acima: a pessoa referenciada pelo identificador *r9ronaldo* tem o nome *Ronaldo* e o e-mail *r9ronaldo@realmadrid.com*. O Real Madrid, cujo Website é *http://www.realmadrid.com/*, tem este atleta em sua equipa.

Usando a sintaxe de RDF/XML, tem-se:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3   xmlns:rm="http://www.realmadrid.com/">
4   <rdf:Description rdf:about="http://www.realmadrid.com/">
5     <rm:Atleta>
6       <rdf:Description rdf:about="http://www.r9ronaldo.com/">
7         <rdf:type rdf:resource="http://http://www.realmadrid.com/Atleta"/>
8         <rm:Nome>Ronaldo</rm:Nome>
9         <rm:Email>r9ronaldo@realmadrid.com</rm:Email>
10      </rdf:Description>
11    </rm:Atleta>
12  </rdf:Description>
13 </rdf:RDF>

```

Observe que no exemplo acima, o prefixo *rm* refere-se a um *namespace* específico definido na declaração XML (linha 3). Um *namespace* (Bray, Hollander, and Layman, 1999) é uma colecção de nomes, definindo uma única vez num documento XML, identificado por uma referência URI, que é posteriormente usadas em outros documentos XML como nomes de elementos ou atributos.

A sintaxe RDF/XML permite que se faça relacionamento entre dados. Faz-se necessário, então, um nível de esquema onde é declarada a existência de novas propriedades. Em

RDF, o RDF-Schema (Brickley and Guha, 2000a) permite verificar, por exemplo, que a carta de condução de um motorista tem o nome de uma pessoa, e não um modelo de carro.

Mais detalhes sobre RDF podem ser encontrados em (Lassila and Swick, 1999; Powers, 2003; Brickley, 2000; Ricker, 2000) e (Ahmed et al., 2001).

### 2.4.2 RDF Schema – RDF(S)

Conforme apresentado acima, RDF foi concebido para descrever metadados sobre recursos. RDF divide-se em duas partes:

1. RDF define como descrever recursos em termos de suas propriedades e valores;
2. RDF Schema define propriedades específicas que podem ser utilizadas para definir esquemas.

Essas duas definições juntas costumam ser referidas como RDF(S) (Staab et al., 2000).

RDF Schema é um sistema de classes extensível e genérico que pode ser utilizado como base para esquemas de um domínio específico. Esses esquemas podem ser compartilhados e estendidos através de refinamento de subclasses. Além disso, definições de metadados podem ser reutilizadas através do compartilhamento de esquemas.

O vocabulário de RDF está definido em dois namespaces: *rdf* e *rdfs*. Declarações RDF devem ser precedidas de um dos dois prefixos abaixo descritos:

- 1 | *rdfs*=<http://www.w3.org/2000/01/rdf-schema#>
- 2 | *rdf*=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

RDF Schema (instâncias de modelos de dados RDF) pode ser visto como diagramas Entidade-Relacionamento (DER), já que as propriedades da RDF representam relacionamentos entre recursos. RDF Schema (Brickley and Guha, 2000b) é responsável por fornecer mecanismos para declaração dessas propriedades. Um esquema não define somente as propriedades dos recursos, mas também os tipos de recursos que estão sendo descritos. Pode ser entendido como uma espécie de dicionário onde são definidos os termos que serão utilizados em declarações RDF.

A linguagem RDF Schema fornece os mecanismos necessários à definição de elementos, de classes de recursos, de possíveis restrições de classes e relacionamentos e detecção de violação de restrições.

Como tudo em RDF é considerado um recurso, RDF Schema estabelece que esses recursos podem ser organizados em classes. Um recurso pode ser instância de uma ou mais classes. A propriedade *rdf:type* é utilizada para indicar as classes das quais um recurso é instância.

As classes podem estar organizadas em uma hierarquia. Isso significa que qualquer recurso de um tipo que é subclasse de outro, é também considerado como sendo do tipo da superclasse. Tal relacionamento entre classes é denotado através da propriedade *rdfs:subClassOf*.

O sistema de tipos de RDF Schema é semelhante ao sistema de tipos do modelo de classes de UML (Fowler and Scott, 2000; OMG, 2003), com algumas pequenas diferenças. Uma delas é que, ao invés de definir classes em termos das propriedades que suas instâncias devem ter, um RDF Schema define propriedades em termos das classes de recursos aos quais elas se aplicam. Este é o papel de *rdfs:domain* e *rdfs:range*. Por exemplo, pode-se definir que a propriedade *professor* possui um domínio *aula* e um valor *string*, enquanto em UML seria definida uma classe *aula* com um atributo chamado *professor* do tipo *string* (Brickley and Guha, 2000a). A especificação de RDF Schema não define nenhum tipo específico de dados, mas permite que eles sejam usados como valor da propriedade *rdfs:range*.

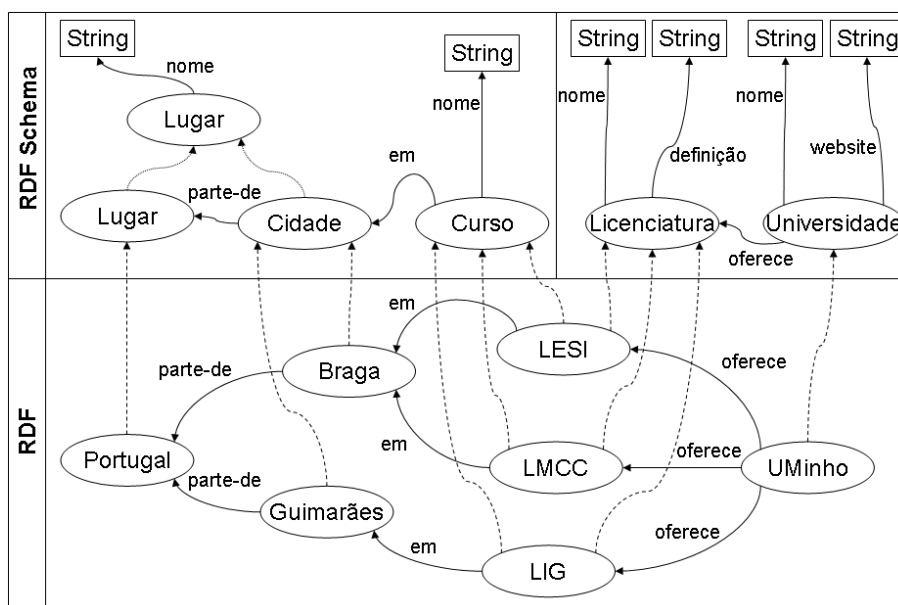


Figura 2.10: Um exemplo envolvendo RDF e RDF Schema

Para fornecer diferentes visões sobre os mesmos recursos, RDF(S) também auxilia a definir visões integradas de recursos heterogêneos. Por exemplo, a Figura 2.10 cita as licenciaturas da Universidade do Minho que são relacionadas com computação. Tais licenciaturas possuem características distintas entre si (como nome e definição), estando também situadas em diferentes cidades. Além disso, cada licenciatura tem seu tempo total de duração do curso.

As camadas RDF e RDF Schema apresentadas na Figura 2.10 fornecem uma solução para a estruturação e representação destes dados.

Outro exemplo de aplicação utilizando o RDF Schema pode descrever a classe *Veículo*, a qual possui as sub-classes *Veículo de Passageiro* e *Veículo de Carga*, conforme mostra a Figura 2.11.

A codificação deste exemplo na sintaxe RDF Schema está abaixo representada:

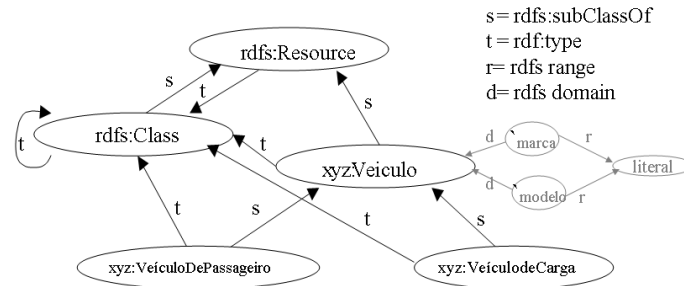


Figura 2.11: Exemplo de aplicação de RDF Schema.

```

1 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22/rdf-syntax-ns#"
2   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
3   <rdf:Description ID="Veiculo">
4     <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
5     <rdfs:subClassOf resource="http://www.w3.org/TR/rdf-schema#Resource"/>
6   </rdf:Description>
7   <rdf:Description ID="VeiculoDePassageiro">
8     <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
9     <rdfs:subClassOf resource="#Veiculo"/>
10  </rdf:Description>
11  <rdf:Description ID="VeiculoCarga">
12    <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
13    <rdfs:subClassOf resource="#Veiculo"/>
14  </rdf:Description>
15 </rdf>

```

Detalhes mais aprofundados sobre RDF Schema podem ser encontrados em (Brickley and Guha, 2000a; Fikes and McGuinness, 2001; Brickley and Guha, 2000b; Powers, 2003) e (Brickley, 2000).

### 2.4.3 Topic Maps – TM

Topic Maps tem por objectivo tornar as estruturas de representação de informação processáveis por ferramentas e passíveis de navegação. Além disso, Topic Maps também fornece técnicas avançadas para a conexão e endereçamento da estrutura da informação com os recursos de informação, tal como um índice.

Tecnicamente falando, Topic Maps (Biezunsky, Bryan, and Newcomb, 1999) é um formalismo para representar a estrutura de um conjunto de recursos de informação, organizando-o em *tópicos*. Esses tópicos têm ocorrências e associações que representam e definem os relacionamentos entre os tópicos, dando-lhes significado. O formalismo de Topic Maps suporta um paradigma que permite organizar, manter e navegar pela informação, de modo a extrair dela conhecimento.

Topic Maps são compostos basicamente por 3 partes simples, porém com alto poder descritivo:

**Tópicos:** pode representar qualquer coisa: uma pessoa, uma entidade, um conceito, etc.

Em rigor, o termo “tópico” refere-se a um elemento do topic map que representa um tema;

**Associações:** permitem ligar dois ou mais tópicos, definindo um relacionamento semântico entre os temas por eles representados;

**Ocorrências:** representa um ou mais recursos de informação que são especificados como relevante para um dado tópico.

Topic Maps pode ser definido como uma descrição de um ponto de vista sobre uma colecção de recursos, organizado formalmente por tópicos, e pela ligação de partes relevantes do conjunto de informação aos tópicos apropriados (Pepper, 2000).

Um topic map é um mapa que representa o conhecimento encontrado em um domínio. Ele apresenta os conceitos relevantes e os relacionamentos entre eles, em um modo similar ao que faz um thesaurus. Ele também dá a definição de conceitos, assim como um dicionário. Topic Maps auxilia a organização dos conceitos em uma taxonomia e uma ontologia<sup>17</sup>.

Permitindo criar um mapa virtual da informação, os recursos de informação mantêm-se em sua forma original e não são modificados. Então, o mesmo recurso de informação pode ser usado de diferentes formas, por diferentes mapas de tópicos. Como é possível e fácil modificar um mapa, a reutilização da informação é conquistada.

A norma Topic Maps é descrita em detalhe no Capítulo 3.

#### 2.4.4 XML-based Ontology Exchange Language – XOL

A linguagem *XML-based Ontology Exchange Layer* (XOL) fornece um formato que permite o intercâmbio de definições contidas em um conjunto de ontologias que se relacionam (Karp, Chaudhri, and Thomere, 1999).

A sintaxe da XOL baseia-se em XML por ser razoavelmente simples de ser validada. A semântica de XOL baseia-se em OKBC-Lite<sup>18</sup> (*Open Knowledge Base Connectivity*), um modelo de conhecimento simplificado.

A estrutura da sintaxe XOL possui 4 partes principais:

- cabeçalho (nome, versão);
- classes/subclasses;
- slots (propriedades);
- instâncias.

---

<sup>17</sup>Sendo esta capacidade aquela que mais interessa ao contexto desta dissertação, discutir-se-á no Capítulo 4 como os topic maps podem descrever ontologias e até como podem ser gerados automaticamente a partir delas.

<sup>18</sup><http://www.ai.sri.com/~okbc/>



Um exemplo simples de uma ontologia representada em XOL é apresentado na Figura 2.12. A classe *Pessoa* possui duas sub-classes, nomeadamente *Homem* e *Mulher*. A classe *Pessoa* possui dois tipos de relacionamentos entre seus membros: *tem pai* e *tem irmão*.

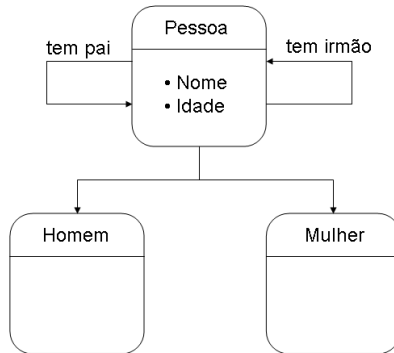


Figura 2.12: Exemplo de ontologia em XOL.

De acordo a Figura 2.12, um exemplo de ontologia representado na sintaxe XOL seria assim descrita:

```

1 <module>
2   <name>Familia - ontologia</name>
3   <version>1.2</versao>
4   <documentation>Esse exemplo visa mostrar a sintaxe XOL.</documentation>
5   <class>
6     <name>Homem</name>
7     <documentation>Classe de pessoas de sexo masculino.</documentation>
8     <subclass-of>Pessoa</subclass-of>
9   </class>
10  <class>
11    <name>Mulher</name>
12    <documentation>Classe de pessoas de sexo feminino.</documentation>
13    <subclass-of>Pessoa</subclass-of>
14  </class>
15  <slot>
16    <name>tem_irmaos</name>
17    <documentation>...</documentation>
18    <domain>Pessoa</domain>
19    <slot-value-type>Homem</slot-value-type>
20  </slot>
21  <slot>
22    <name>tem_pai</name>
23    <documentation>...</documentation>
24    <domain>Pessoa</domain>
25    <slot-value-type>Homem</slot-value-type>
26    <slot-inverse>pai_de</slot-inverse>
27  </slot>
28  <individual>
29    <name>Pedro</name>
30    <instance-of>Homem</instance-of>
31    <slot-values>
32      <name>tem_pai</name>
33      <value>Paulo</value>
34    </slot-values>
35  </individual>
36  <individual>
37    <name>Paulo</name>

```

```

38 |         <instance-of>Homem</instance-of>
39 |         <slot-values>
40 |             <name>pai-de</name>
41 |             <value>Pedro</value>
42 |         </slot-values>
43 |     </individual>
44 | </module>

```

As características da ontologia estão descritas no princípio (linhas 2–4). Nos elementos `<class>` encontram-se as classes desta ontologia – homem e mulher (linhas 5–14). Nos elementos `<slot>` declaram-se as propriedades dos relacionamentos (linhas 15–27). Por fim, os elementos `<individual>` apresentam as instâncias desta ontologia.

O que vale observar neste exemplo são os relacionamentos inversos: *Pedro diz que seu pai é Paulo* (linhas 32–33), enquanto *Paulo diz que é pai de Pedro* (linhas 40–41).

#### 2.4.5 Simple HTML Ontology Extensions – SHOE

SHOE (Luke and Helfin, 2000) é uma linguagem de representação de conhecimento baseada em HTML que adiciona etiquetas necessárias para a representação semântica de dados em páginas Web. SHOE é uma extensão do HTML que permite:

- definição de ontologias;
- definição de regras (através de cláusulas Horn);
- fazer anotações em páginas HTML a partir de propriedades de uma ou mais ontologias.

As etiquetas de SHOE estão divididas em duas categorias:

1. Etiquetas para construir ontologias. As ontologias em SHOE são conjuntos de regras que definem que tipo de declarações podem fazer os documentos SHOE e o que significam estas declarações. Por exemplo, uma ontologia SHOE pode decidir que uma entidade de dados é um *automóvel* e isso significa que *automóvel* pode possuir um *nome*;
2. etiquetas que permitam anotar documentos Web que pertençam a uma ou mais ontologias, declarar entidades e criar declarações sobre essas entidades sob as regras estabelecidas pelas ontologias. Por exemplo, um documento SHOE que se pertença à ontologia descrita anteriormente pode declarar um automóvel chamado *Audi A3*.

SHOE oferece uma base de conhecimento semântico, pois foi projectada tendo em mente as necessidades da busca de informação na Web. Ela possui várias formas de poder gerir os dados que se distribuem e está orientada principalmente a facilitar o trabalho dos agentes da Web e aos agentes inteligentes de maneira que possam tornar as pesquisas na Web mais completas. Entretanto, possui uma semântica limitada.

Um exemplo simples de como criar axiomas em SHOE é visto a seguir:

```

1 <DEF-INFERENCE DESCRIPTION= "irmaos(?pes1,?pes2) if tem pai (pes1,?pessoa) and tem pai(pes2,?pessoa)">
2   <INF-IF>
3     <RELATION NAME="tem pai">
4       <ARG POS=1 VALUE="pes1" USAGE=VAR>
5       <ARG POS=2 VALUE="pessoa" USAGE=VAR>
6     </RELATION>
7     <RELATION NAME="tem pai">
8       <ARG POS=1 VALUE="pes2" USAGE=VAR>
9       <ARG POS=2 VALUE="pessoa" USAGE=VAR>
10    </RELATION>
11  </INF-IF>
12  <INF-THEN>
13    <RELATION NAME="irmaos">
14      <ARG POS=1 VALUE="pes1" USAGE=VAR>
15      <ARG POS=2 VALUE="pes2" USAGE=VAR>
16    </RELATION>
17  </INF-THEN>
18 </DEF-INFERENCE>

```

O exemplo acima apresenta a definição de uma regra que define se duas pessoas são irmãos: será sempre verdade se ambas possuírem o mesmo pai.

#### 2.4.6 Web Ontology Language – OWL

OWL (*Web Ontology Language*) (Patel-Schneider et al., 2004) é um trabalho em andamento no W3C, desde Março de 2002, que faz parte da actual lista de recomendações da W3C para o desenvolvimento da *Semantic Web* (Patel-Schneider et al., 2004):

**XML:** fornece a sintaxe para descrever a estrutura de documentos, mas não dispõe de uma componente semântica que dê significado aos documentos;

**XML Schema:** é uma linguagem para definir a sintaxe e ainda restrições relacionadas com estruturas de documentos XML;

**RDF:** é um modelo de dados para objectos (recursos) e relações existentes entre eles, fornecendo uma semântica simples para esse modelo, o qual pode ser representado através de uma sintaxe XML;

**RDF Schema:** é uma notação para descrever as propriedades e classes dos recursos de RDF, com uma semântica referente a hierarquias e generalizações de tais classes e propriedades;

**OWL:** completa a linguagem anterior, adicionando-lhe construções para a descrição das classes e propriedades como, por exemplo, relações entre estas classes, cardinalidades, igualdades, tipos e características mais apurados; ou seja, permite definir um maior número de restrições semânticas que a rede de recursos de informação deve respeitar.

OWL é utilizada para representar explicitamente o conjunto de termos de um vocabulário e os relacionamentos entre estes termos. OWL é uma linguagem para a descrição de ontologias desenvolvida para o uso de aplicações para a Web que necessitam processar o

conteúdo de recursos de informação disponíveis na Web, em vez de somente apresentá-las aos utilizadores; ou seja, OWL é utilizada quando as informações contidas em documentos precisam ser processadas por aplicações, ao contrário do que ocorre quando estes conteúdos apenas são mostrados aos utilizadores humanos que então procedem à sua interpretação.

OWL possui três sub-linguagens, que foram projectadas conforme o grupo de implementadores e utilizadores: OWL Lite, OWL DL e OWL Full (McGuinness and van Harmelen, 2004).

**OWL Lite:** hierarquia e características simples de restrição. Ex: valores de cardinalidade limitados a 0 e 1;

**OWL DL:** maior expressividade sem perda de poder computacional e capacidade de decisão dos sistemas. Inclui todos os construtores do OWL;

**OWL Full:** nível máximo de expressividade e liberdade sintáctica do RDF. Ex: uma classe pode ser tratada simultaneamente como uma colecção de elementos individuais e como um elemento individual.

OWL oferece uma maior facilidade para expressar a semântica que XML, RDF e RDF Schema, pois supera estas linguagens em sua habilidade em representar, de forma legível, o conteúdo na Web. OIL e DAML iniciaram como projectos de pesquisa, e com o tempo fundiram-se em DAML+OIL. Esta linguagem foi adaptada pela W3C e normalizada como OWL, a qual sucede às anteriores. Portanto, OWL é uma revisão das linguagens DAML (*DARPA Agent Markup Language*) e OIL (*Ontology Inference Layer*), acrescentando-lhes características aprendidas no desenvolvimento e nas aplicações da DAML+OIL.

A sintaxe da linguagem OWL para definição de uma ontologia inicia com uma declaração, em RDF, contendo os *namespaces* que serão utilizados na ontologia. Basicamente, a ontologia começa da forma que se ilustra no exemplo seguinte (Bechhofer et al., 2004):

```

1 <rdf:RDF
2   xmlns="http://www.w3.org/2001/sw/WebOnt/guide-src/wine#"
3   xmlns:vin="http://www.w3.org/2001/sw/WebOnt/guide-src/wine#"
4   xmlns:food="http://www.w3.org/2001/sw/WebOnt/guide-src/food#"
5   xmlns:owl="http://www.w3.org/2002/07/owl#"
6   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
7   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
8   xmlns:xsd="http://www.w3.org/2000/10/XMLSchema#">

```

As três primeiras declarações indicam o *namespace* associado à ontologia que está sendo desenvolvida que, no exemplo mostrado, se refere a vinhos e comidas. Na linha 5, é indicado o *namespace* específico da OWL e nas linhas subsequentes têm-se as referências tradicionais aos vocabulários específicos de RDF, RDF Schema e XML Schema, respectivamente.

Uma vez que os *namespaces* estão definidos, inclui-se, através do elemento <owl:Ontology>, informações como comentários, controle de versões e inclusão de outras ontologias, caso esteja havendo extensão de uma já existente. Abaixo apresenta-se a título de exemplo um

fragmento do código OWL referente a estas informações (Smith, Welty, and McGuinness, 2004):

```

1 <owl:Ontology rdf:about="">
2   <rdfs:comment>Um exemplo de ontologia OWL</rdfs:comment>
3   <owl:priorVersion
4     rdf:resource="http://www.w3.org/2001/sw/WebOnt/guide-src/wine-112102.owl"/>
5   <owl:imports rdf:resource="http://www.w3.org/2001/sw/WebOnt/guide-src/food.owl"/>
6   ...

```

Têm-se no código acima a representação de um cabeçalho OWL contendo informações diversas representadas por:

- *rdfs:comment* apresenta algum comentário referente à ontologia modelada;
- *owl:priorVersion* é um elemento utilizado para o controle de versões da ontologia;
- *owl:imports* indica a importação de uma ontologia já existente definida pelo recurso apresentado em seguida.

As classes e subclasses OWL são definidas, respectivamente, através dos elementos *owl:Class* e *rdfs:subClassOf*. Por exemplo, a declaração de classe e sub-classe em OWL é realizada da seguinte forma:

```

1 <owl:Class rdf:ID="Curso"/>
2 <owl:Class rdf:ID="Licenciatura">
3   <rdfs:subClassOf rdf:resource="#Curso"/>
4   ...
5 </owl:Class>

```

No código OWL acima, mostra-se a declaração de duas classes: *Curso* (linha 1) e *Licenciatura* (linha 2 a 5). Sendo que, na linha 3, é especificado que *Licenciatura* é uma sub-classe de *Curso*.

Através do elemento *owl:ObjectProperty* define-se as propriedades das classes e subclasses. Utiliza-se – em conjunto com o elemento *<ObjectProperty>* – o elemento *rdfs:domain* para especificar a qual classe a referida propriedade está sendo atribuída. Abaixo, segue um simples exemplo da declaração de propriedades em OWL:

```

1 <owl:ObjectProperty rdf:ID="nome"/>
2   <rdfs:domain rdf:resource="#Curso"/>
3 </owl:ObjectProperty>

```

A propriedade *nome* está declarada na primeira linha e é pertencente à classe *Curso*, de acordo com a linha 2.

A OWL oferece vários elementos como *owl:Restriction*, *owl:cardinality*, *owl:minCardinality*, *owl:maxCardinality*, *owl:inverseOf*, entre diversos outros, os quais permitem determinar, respectivamente, as restrições, cardinalidades, cardinalidade mínima, cardinalidade

máxima, relação inversa, de modo a obter uma ontologia mais completa e melhor modelada.

Para maiores detalhes sobre OWL, recomenda-se a leitura de (Bechhofer et al., 2002; Smith, Welty, and McGuinness, 2004; McGuinness and van Harmelen, 2004; Bechhofer et al., 2004; Patel-Schneider et al., 2004; Daconta, Obrst, and Smith, 2003; Fensel, 2002; Coakes, 2003).

### 2.4.7 Comparativo entre as linguagens

A comparação entre as linguagens até o momento apresentadas pode ser visualizada na Tabela 2.1, a qual foi estendida de (Su and Ilebrikke, 2002) e (Gómez-Pérez and Corcho, 2002).

Tabela 2.1: Comparativo entre as Linguagens para *Semantic Web*

	RDF(S)	OIL	OWL	XOL	SHOE	TM
(A) Poder de Expressividade	média -	média -	média +	média	média	média +
(B) Número de Construtores	pequeno	pequeno	médio -	médio	pequeno	médio
(C) Mecanismos de abstracção	cla, gen agre, ass	cla, gen agre, ass	cla, gen agre, ass	cla, gen agre, ass	cla, gen agre, ass	cla, gen agre, ass
(D) Axiomas	não	sim	sim	não	sim	não
(E) Inferência	não	sim	sim	não	sim	não
(F) Mecanismo de restrições	fraca	fraca	fraca	fraca	muito fraca	fraca -

Legenda da Tabela 2.1:

- (A) “Média -”: conjunto limitado de propriedades;
- (B) “Agre”: somente entre classes; disjunção apenas em OIL, OWL e TM;
- (D) Tem como base Lógica de Primeira Ordem: OIL e OWL permitem definir propriedades algébricas de relações (transitivas, simetria, negação, etc.);
- (E) “Fraca -”: as associações permitem especificar um conjunto de restrições de um tipo.

Actualmente, não há um motor de inferência lógica existente que suporte Topic Maps, porém há algumas formas de se conseguir esta inferência. A linguagem de interrogação *tolog* (Garshol, 2004e) pode efectuar inferências lógicas usando uma abordagem similar a Datalog (Liu and Cleary, 1996; Ullman and Zaniolo, 1990; Elmasri and Navathe, 1994); no momento, há ao menos duas implementações disso (TM4J (SourceForge, 2003b) e OKS (Ontopia, 2004)).

É importante apontar que Topic Maps não foram projectados para a inferência lógica. Originalmente, seu propósito foi auxiliar na recuperação avançada de informação, embora sua área de utilização tem sido expandida. Isto significa que inferência lógica é apenas uma das muitas coisas que se podem fazer com Topic Maps.

No problema das restrições, o Capítulo 9 se dedica em apresentar uma abordagem que permite definir regras sobre Topic Maps, melhorando assim sua posição na comparação com as outras linguagens para *Semantic Web*.

## 2.5 Sumário do capítulo

Esse capítulo teve por objectivo introduzir os principais conceitos que serão vistos ao longo desta dissertação. Primeiramente, as definições de dado, informação e conhecimento foram clarificadas, de modo a evitar ambiguidade nos próximos capítulos.

A Secção 2.2 classificou e descreveu algumas das principais representações de conhecimento utilizadas pela comunidade académica, sempre tentando manter um relacionamento entre estes métodos de representação.

*Semantic Web* – uma área na qual encaixa-se esta dissertação – foi o tema da Secção 2.3. A *Semantic Web* é vista como uma nova geração da Web actual, a qual tem como finalidade atribuir um significado aos conteúdos publicados na Web de modo que seja perceptível tanto pelo humano como pelo computador. Para isto, as informações devem estar representadas de alguma forma que se possa extrair o conhecimento nelas inserido.

A fim de permitir uma codificação do conhecimento a ser representado na *Semantic Web*, a Secção 2.4 descreveu as principais linguagens para a especificação de ontologias, thesauri e taxonomias, no âmbito da *Semantic Web*.

A partir do levantamento bibliográfico realizado até este ponto, o próximo capítulo apresenta Topic Maps como a nossa escolha para a representação do conhecimento, justificando o porque desta decisão.





## Capítulo 3

# Topic Maps

*Eles não sabem nem sonham  
Que o sonho comanda a vida  
Que cada vez que o homem sonha  
O mundo pula e avança  
Como bola colorida  
Nas mãos de uma criança*

Antonio Gedeão

Conforme a ISO (*International Organization for Standardization*) (ISO, 2005) e a IEC (*International Electrotechnical Commission*) (IEC, 2005), **Topic Maps (TM)** é uma norma internacional unificada para descrever estruturas de conhecimento e formalizar a sua associação com recursos de informações. Como tal TM é uma tecnologia habilitada para a representação e manipulação do conhecimento, propiciando também uma poderosa forma de navegação sobre recursos de informação.

Os primeiros trabalhos no conceito dos **Topic Maps** datam de princípios de 1991. Um grupo de pessoas, chamado *Davenport group*, decidiu dar início ao desenvolvimento de uma norma para documentação de software. Entretanto este grupo dividiu-se em dois: *DocBook* (Walsh and Muellner, 1999) e *CapH* (*Conventions for the Application of HyTime*) (Rath and Pepper, 1999). *HyTime* (Newcomb, Biezunski, and Bryan, 2003) foi concebida nesta ocasião.

Com a participação de Fred Dalrymple, Michel Biezunski, Wayne Wholer, entre outros, o *CapH* continuou o desenvolvimento de diferentes modos para modelar aquilo a que chamaram **Topic Maps**. Depois de criarem este conceito, Michel Biezunski prosseguiu com o seu desenvolvimento, quase sozinho, durante vários anos.

Entretanto, uma tentativa de criação de uma norma chamado *Topic Navigation Maps*, definida por Michel Biezunski, foi efectuada e aceite pelo grupo ISO SGML em 1996, sendo então criado um projecto para um nova norma internacional – ISO/IEC 13250 (Biezunsky, Bryan, and Newcomb, 1999). Martin Bryan e Steve Newcomb juntaram-se a Michel

Biezunski em 1997 e 1998 respectivamente. Depois de intensas discussões, a norma foi finalmente submetida aos membros da ISO em Outubro de 1998.

Durante quatro meses registraram-se ainda várias alterações. O resultado final é um compromisso que se pensa ser de extremo poder expressivo e flexibilidade por um lado, e de uma semântica suficientemente bem definida pelo outro (Biezunski and Newcomb, 1999). O modelo, inicialmente concebido para a união de índices, foi desenvolvido para algo muito mais poderoso e não restrito apenas a esse fim.

Básica e informalmente, um topic map é um grafo em que os vértices são tópicos (conceitos) de um determinado domínio e os arcos são qualquer ligação existente entre os tópicos, formando assim uma rede semântica (mapa) de conceitos.

A disponibilização de informação cria a necessidade de organização interna e meta-informação de tal forma que os motores de busca possam entregar melhores resultados quando interrogados sobre um tema em particular.

A publicação da norma **Topic Maps** atraiu muita atenção das comunidades de representação e gestão de conhecimento e de *Semantic Web*.

Mack (Mack and Yael Ravin, 2001) afirmou que a gestão de conhecimento refere-se aos métodos e ferramentas para capturar, armazenar, organizar e tornar o conhecimento acessível. Esta área de gestão do conhecimento, de acordo com Sigel (Sigel, 2000), está interessada em otimizar a organização dos repositórios de conhecimento para suportar uma fácil criação, recuperação e partilha do conhecimento entre os consumidores. No momento em que há um crescimento exponencial da informação, **Topic Maps** introduzem uma estrutura de conhecimento que é processável por máquina e permite a navegação.

**Topic Maps** também tem um papel potencial no campo de *Semantic Web*. *Semantic Web* é uma extensão da Web corrente na qual a informação é dada em um sentido bem-definido, possibilitando que computadores e humanos possam trabalhar em cooperação (Berners-Lee, Hendler, and Lassila, 2001). Essa cooperação é realizada por agentes inteligentes que possuem a capacidade de interpretar o significado

Embora **Topic Maps** não incluam agentes inteligentes – os quais fornecem aos computadores a capacidade de interpretar o significado contido na *Semantic Web* – outros aspectos desta tecnologia se adequam neste contexto. Através de *Topic Map Query Language* (TMQL) (Garshol and Barta, 2005), a qual encontra-se em desenvolvimento – os requisitos da linguagem estão especificados em (Garshol and Barta, 2003) – os recursos dentro de um topic map podem ser recuperados consequentemente.

O conceito **Topic Maps** pode ser definido como uma descrição de um ponto de vista sobre uma colecção de recursos, organizado formalmente por tópicos e pela ligação de partes relevantes do conjunto de informação aos tópicos apropriados. Um mapa de tópicos expressa a opinião de alguém sobre o que os tópicos são, e quais as partes do conjunto de informação que são relevantes para cada tópico. Falar sobre **Topic Maps** é falar sobre estrutura de conhecimento.

Os principais objectivos de **Topic Maps** são (Rath, 2003):

- Estruturar recursos de informação não estruturados, com mecanismos externos aos recursos;
- Permitir procuras que recuperem a informação requisitada;
- Criar visões diferentes para utilizadores ou finalidades específicas, filtrando a informação.

Permitindo criar um mapa virtual da informação, os recursos de informação mantêm-se em sua forma original e não são modificados. Então, o mesmo recurso de informação pode ser usado de diferentes formas, por diferentes mapas de tópicos. Como é possível e fácil modificar um mapa, a reutilização da informação é conquistada.

### 3.1 A norma ISO 13250 Topic Maps

Como *tema* é uma das palavras-chaves de **Topic Maps**, começamos por apresentar sua definição.

Tema (*subject*) é um conceito, uma noção, uma ideia. De acordo com o dicionário de Cambridge<sup>1</sup>, *subject* é a coisa que está sendo estudada, discutida e/ou considerada. Na norma **Topic Maps**, a definição de tema é (Biezunsky, Bryan, and Newcomb, 1999):

Em um sentido genérico, um tema é qualquer coisa, o que quer que seja, sem levar em consideração se esta coisa existe ou pode ser concretizada e individualizada de qualquer forma, sobre a qual é possível fazer asserções e raciocínio.

A generalidade dessa definição permite que **Topic Maps** possa ser aplicado em qualquer domínio de aplicação que se possa pensar (Rath, 2003).

Qualquer tema relevante dentro do domínio de aplicação será mapeado em um tópico. Para isso, este mapeamento depende do:

**Domínio da Aplicação:** define o universo (conjunto fechado de “coisas”) a que o topic map se refere, estabelecendo, por conseguinte, o conjunto dos temas incluídos. Apesar do conhecimento ser infinito, o domínio da aplicação deve ser limitado, para ser gerido;

**Utilizador do topic map:** geralmente, um topic map é construído para utilizadores. Consequentemente, seus requisitos devem ser cobertos e esses mesmos requisitos podem diferenciar-se, dependendo da perspectiva do utilizador, do contexto, da visão ou do conhecimento sobre o domínio da aplicação.

---

<sup>1</sup><http://dictionary.cambridge.org/>

**Autor do topic map:** um topic map é um resultado de um processo intelectual realizado por um autor. Este autor é o responsável por definir o que é o mapa, isto é, quais os temas (tópicos) a incluir e como os relacionar. As selecções feitas dependem de seu conhecimento individual sobre o domínio da aplicação e dos requisitos do utilizador.

Para exemplificar, mostra-se alguns temas que demonstram que os mesmos dependem do público destino. Neste caso, usaremos *futebol* como domínio:

- Temas sobre *atleta*: clube, origem, contrato, histórico de clubes, títulos, convocações, salário, ...
- Temas sobre *campeonato*: regras, estádio, campeão, clube, artilheiro, federação, ...
- Temas sobre *dirigente*: federação, clube, profissão, reunião, salário, aquisição, ...

Como já mencionado, temas resultam de um processo intelectual realizado pelo autor do topic map para modelar recursos de informação e conhecimentos de pessoas, podendo ser partes concretas ou conceitos abstractos do mundo a ser modelado. E, geralmente, os temas estão situados fora do domínio dos computadores – o computador não conhece os temas. Por exemplo, um estádio de futebol, como o Beira-Rio (do Inter-RS), está fisicamente situado em Porto Alegre - Brasil, portanto não podendo ser acedido directamente por algum computador.

Introduzida a definição básica de **Topic Maps**, há necessidade de encontrar uma notação precisa e universal para o descrever. Há de momento várias alternativas plausíveis, tais como XTM – XML Topic Maps (Pepper and Moore, 2001a), HyTM (Newcomb, Biezunski, and Bryan, 2003), AsTMa= (Barta, 2004), ou LTM (Garshol, 2002c), que serão detalhadas à frente na Secção 3.4. Destas apenas a XTM é um standard (as outras foram criadas por distintos fabricantes para os seus sistemas específicos) – HyTM foi a sintaxe original para a norma **Topic Maps**, porém foi substituída pelo XTM. Por isso, e porque assenta em XML, essa tem sido a sintaxe adoptada em muitos projectos e vai ser a escolhida na presente dissertação. Assim, e antes de prosseguir com a apresentação da norma suportada no triângulo conceptual TAO, far-se-á uma introdução do XTM que é usado nos exemplos.

### 3.1.1 XML Topic Maps (XTM)

**Topic Maps** podem ser expressados usando XML. Para isto, um grupo de pesquisadores, liderados por Steve Pepper e Graham Moore, definiu a linguagem XTM (*XML Topic Maps*) (Pepper and Moore, 2001a), criando o *TopicMaps.org*. *TopicMaps.Org*<sup>2</sup> é uma associação independente de grupos de desenvolvimento, projectando a aplicabilidade da norma **Topic Maps** para a internet com o uso das características da família de especificação XML. Portanto, XTM é o formato para intercâmbio (interoperabilidade) de **Topic Maps** entre aplicações.

---

<sup>2</sup><http://www.topicmaps.org>

XTM é um dialecto XML criado formalmente para descrever todos os conceitos relativos a **Topic Maps** introduzidos pela norma ISO 13250 (Biezunsky, Bryan, and Newcomb, 1999). Assim o respectivo DTD define as notações (*tags*) necessárias e suficientes para identificar os diferentes elementos que constituem um topic map.

Foi decidido adoptar XML como formato de representação de **Topic Maps** devido aos seguintes factores:

- XML é, por excelência, a linguagem actual para a anotação de documentos;
- XML é presentemente a plataforma para intercâmbio de informação mais utilizada;
- XML permite a validação da estrutura sintáctica do topic map;
- XML foi projectado para a representação de dados semi-estruturados, servindo para **Topic Maps** onde os dados estão perfeitamente estruturados;
- XML permite um intercâmbio de dados entre sistemas distintos;
- O processamento de XML pode gerar qualquer formato de saída;
- XML permite que o texto seja compreensível a humanos e computadores, evitando ambiguidades.

A validação sintáctica de um topic map, escrito no formato XTM, é realizada por um qualquer parser XML porque a sua estrutura é regulada por um DTD (definido no documento oficial (Pepper and Moore, 2001b)). Esta definição cria um dialecto XML que contempla todos os conceitos estabelecidos na norma **Topic Maps** (Biezunsky, Bryan, and Newcomb, 1999). A estrutura de um documento XTM está representada na Figura 3.1.

Ou seja, um documento XTM é composto por uma *tag* raiz <topicMap> (a qual inclui os *namespaces*) que pode conter um número qualquer de elementos <topic>, <association> e <mergeMap>, os quais serão apresentados nas próximas subsecções.

### 3.1.2 TAO - Tópicos, Associações e Ocorrências

Desde o princípio, o formalismo para descrição de conhecimento **Topic Maps** assenta-se em três conceitos básicos, designados pela sigla *TAO* (Pepper, 2000): Tópicos, Associações, e Ocorrências. Apesar da simplicidade desde triângulo basilar a abrangência da é tal que a definição possibilita representar estruturas complexas de informação de uma maneira intuitiva.

Nas subsecções seguintes serão apresentados os elementos principais de **Topic Maps** (TAO), os quais estão ilustrados na Figura 3.2.

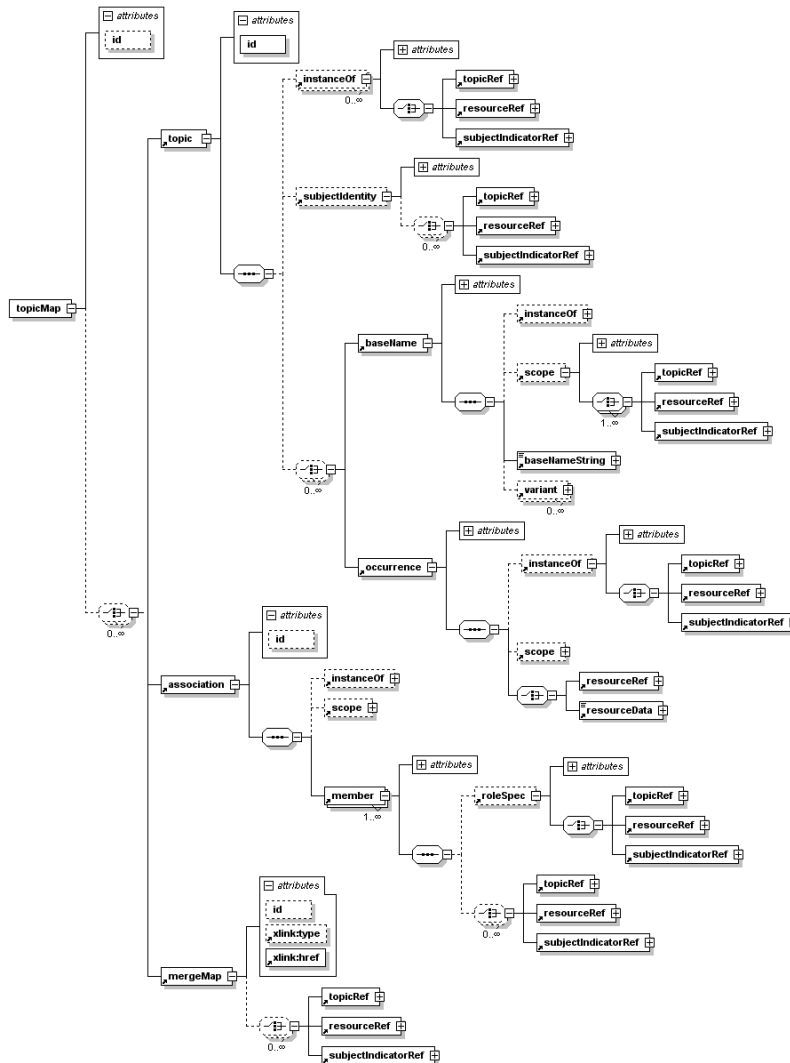


Figura 3.1: Esquema da sintaxe XTM

### Tópicos (*Topics*)

Um tópico em **Topic Maps** representa um tema em um domínio de aplicação. Por exemplo, no domínio *Futebol em Portugal*, podem ser tópicos: *Futebol Clube do Porto*, *Sporting de Braga*, *Sporting Lisboa e Benfica*, *Super Liga*, etc. Em termos técnicos, o relacionamento entre um tópico e um tema é definido como reificação<sup>3</sup>.

Um tópico possui cinco características principais:

**Identificador:** permite que um tópico possa ser diferenciado dos demais tópicos, pois em um topic map o identificador é único;

<sup>3</sup>Do inglês – *reify*: transformar algo abstracto em concreto. Reificação é o processo de criação de um tópico para um tema.

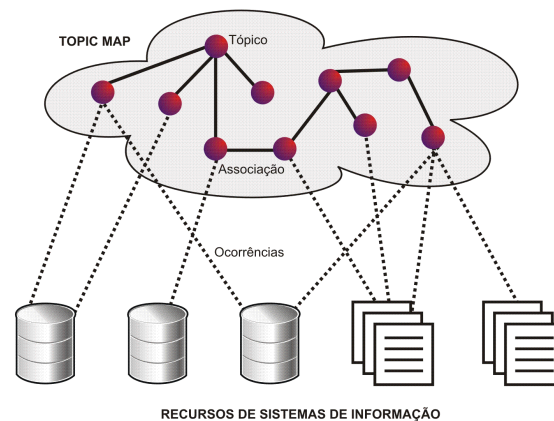


Figura 3.2: Ilustração Simples da Norma Topic Maps

**Tipo:** um tópico pode ser instância de outro tópico, definindo assim uma relação taxonómica;

**Nomes:** nomes (legíveis por humanos) alternativos para designar o conceito/tema em causa;

**Identidade de tema:** indica qual tema do universo de discurso um tópico reifica. Quando dois tópicos tiverem a mesma identidade de tema, considera-se que ambos identificam a mesma coisa, portanto devem ser unidos em um único tópico;

**Ocorrências:** recursos de informação que, embora externos, são relevantes para caracterizar/descrever o tópico;

Para exemplificar os conceitos aqui apresentados, será usado o domínio *Doutoramento em Informática*, onde serão necessários tópicos como os que se seguem para representar temas como: Departamento, Aluno, Professor, Tese, Escola e Universidade, conforme esquematizado na Figura 3.3.

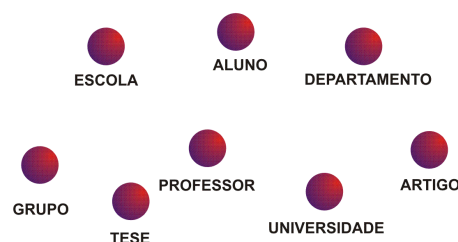


Figura 3.3: Tópicos

Traduzindo a Figura 3.3 para a sintaxe XTM, cada tópico estará declarado em uma *tag* `<topic>`, como exemplificado a seguir:

```
1 | <topic id="aluno">
2 |   <baseName>
```

```

3 |         <baseNameString>Aluno</baseNameString>
4 |     </baseName>
5 | </topic>
6 | <topic id="professor">
7 |     <baseName>
8 |         <baseNameString>Professor</baseNameString>
9 |     </baseName>
10| </topic>
11| ...

```

No código XTM acima, são apresentados dois tópicos: *pessoa* (linha 1) e *professor* (linha 6). Ambos apresentam um nome cada (*baseName*), como característica.

### Tipos de Tópicos (*Topic Types*)

Os tópicos podem ser categorizados dependendo do seu tipo (*topic types*). Em Topic Maps, qualquer tópico pode ser instância de zero ou mais tipos de tópicos, assim *Pedro Rangel Henriques* pode ser mapeado como um tópico do tipo *professor* e *Giovani Rubert Librelotto* como um tópico do tipo *aluno*.

Na Figura 3.4, referente ao universo de discurso de uma universidade, os tipos de tópicos podem ser: aluno, professor, grupo, departamento, artigo, etc.

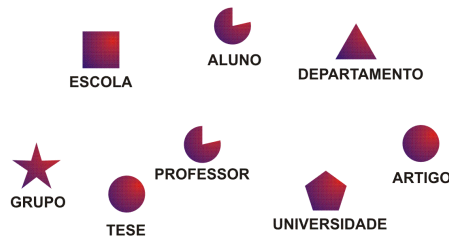


Figura 3.4: Tipos de Tópicos

A relação *tipo de tópico/tópico* pode ser visto como a relação genérica *super-classe/sub-classe*; embora no mundo dos Topic Maps seja encarado como a relação *classe/instância*, na realidade um tópico que é uma instância de um tipo de tópico pode, por sua vez, ser depois instanciado por outro tópico; ou seja, pode ser usado como tipo de tópico. Esta ideia é ilustrada pelo exemplo que se esquematiza na Figura 3.4 e que se encontra descrito na linguagem XTM a seguir.

Um exemplo dessa definição afirma que *aluno* e *professor* são do tipo *pessoa*. Todas as características de *pessoa* são herdadas pelos tópicos instanciados por *aluno* e por *professor*. Uma representação disto encontra-se na Figura 3.5.

A representação da Figura 3.5 na sintaxe XTM, poderia ser como encontrado abaixo:

```

1 | <topic id="pessoa"/>
2 | <topic id="professor">
3 |     <instanceOf>
4 |         <topicRef xlink:href="#pessoa"/>

```



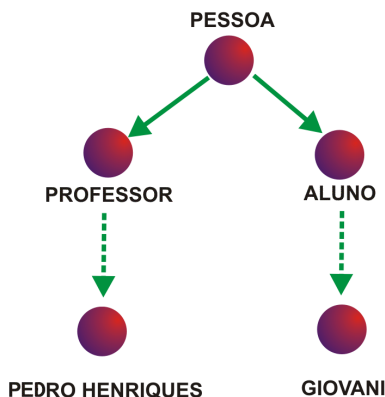


Figura 3.5: Super-classes e sub-classes

```

5      </instanceOf>
6    </topic>
7    <topic id="aluno">
8      <instanceOf>
9        <topicRef xlink:href="#pessoa"/>
10     </instanceOf>
11  </topic>
12  <topic id="pedro-henriques">
13    <instanceOf>
14      <topicRef xlink:href="#professor"/>
15    </instanceOf>
16  </topic>
17  <topic id="giovani-librelotto">
18    <instanceOf>
19      <topicRef xlink:href="#aluno"/>
20    </instanceOf>
21  </topic>

```

Em XTM, as relações *tipo de tópico/tópico* são sempre definidas pelo elemento `<instanceOf>`. Por exemplo: o tipo de tópico de *professor* é uma instância de *pessoa* (linha 4); por sua vez, o tópico de *Pedro Rangel Henriques* é *professor* (linha 14), definindo assim uma hierarquia de três níveis envolvendo os tópicos *pessoa*, *professor* e *Pedro Rangel Henriques*. O mesmo ocorre com os tópicos *pessoa*, *aluno* e *Giovani Rubert Librelotto*.

### Ocorrência (*Occurrence*)

Um tópico pode estar ligado a um ou mais recursos de informação, os quais são relevantes ao tópico em questão de algum modo. Os tais recursos são chamados ocorrências de tópicos. Em termos técnicos, uma ocorrência é uma conexão que relaciona um tópico com um recurso. As ocorrências podem ser de dois tipos:

**Referência ao recurso (*resourceRef*):** Uma ocorrência *resourceRef* conecta o recurso relevante para o tópico usando a notação denominada XLink/XPointer (DeRose, Maler, and Orchard, 2001; DeRose et al., 2002) URI (*Universal Resource Identifier*) (IETF, 1998). Esta referência funciona de forma similar a *hyperlinks* HTML na

web. Consequentemente, todos os recursos endereçáveis por XLink/XPointer URI podem se tornar ocorrências em Topic Maps – significa que tudo o que pode ser endereçado na internet, pode ser ocorrência de tópico;

**Referência a dados (*resourceData*):** Uma ocorrência *resource data* associa um valor (expressado em literal) a um tópico. Isso pode ser usado para associar metadados aos tópicos. A norma Topic Maps não define qualquer tipo de dados para este tipo de recurso. Portanto, a interpretação da sequência de caracteres referente à informação do recurso é dada pela aplicação que interpretará o topic map.

Para exemplificar, considere-se as seguintes ocorrências do tópico que corresponde ao tema *Universidade do Minho*, designação que aqui se usa para identificar o tópico:

- O website da *Universidade do Minho* – recurso documental que tem imensa informação sobre a universidade e que por já existir fisicamente no endereço `http://www.uminho.pt` não vai ser aqui inserido, guardando-se apenas uma referência que será precisamente o URI (ver abaixo linhas 5-10, em particular a linha 9);
- O ano de fundação da *Universidade do Minho* – recurso de informação que, neste caso por se restringir a um único dado (o número 1973), vai ser aqui directamente inserido, escrevendo-se na ocorrência o respectivo valor (ver abaixo linhas 11-17, em particular a linha 15).

Lista abaixo, na sintaxe XTM, o exemplo da declaração do tópico *Universidade do Minho* com as suas duas ocorrências:

```

1 <topic id="uminho">
2   <baseName>
3     <baseNameString>Universidade do Minho</baseNameString>
4   </baseName>
5   <occurrence>
6     <instanceOf>
7       <topicRef xlink:href="#website"/>
8     </instanceOf>
9     <resourceRef xlink:href="http://www.uminho.pt"/>
10  </occurrence>
11  <occurrence>
12    <instanceOf>
13      <topicRef xlink:href="#ano-fundacao"/>
14    </instanceOf>
15    <resourceData>1973</resourceData>
16  </occurrence>
17 </topic>

```

Note-se o efeito deste conceito na separação do topic map em duas camadas: dos tópicos e suas ocorrências. Esta separação é um das vantagens de Topic Maps, conforme visualizado na Figura 3.6.

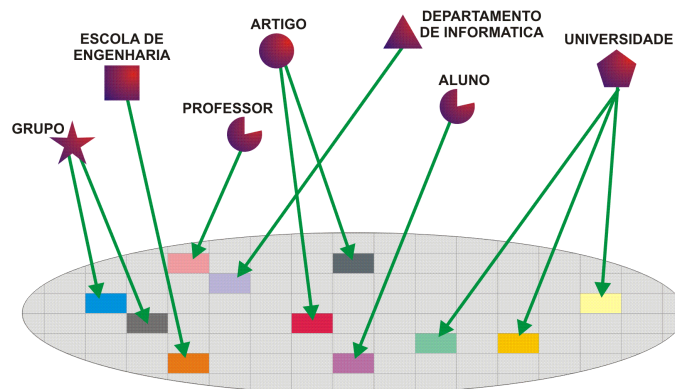


Figura 3.6: Ocorrências de Tópicos

### Associação (*Association*)

Até agora, todos os conceitos discutidos foram referentes a tópicos como o princípio de organização básico para a informação. As definições *tópico*, *tipo de tópico* e *ocorrência* nos permitem organizar os recursos de informação, criando estruturas simples, que apenas individualizam os temas (ou conceitos) do domínio de aplicação; falta, agora, interligá-los.

Uma associação permite descrever relacionamentos entre tópicos. Uma associação é (formalmente) um elemento de vínculo que define um relacionamento entre dois ou mais tópicos. Conforme a Figura 3.7, exemplos de associações podem ser:

- Aluno **escreve** artigos – artigos **são escritos por** alunos;
- Universidade **acolhe** alunos – alunos **estudam na** universidade;
- Professor **pertence a** um grupo – grupos **contém** professores;
- etc...

Para exemplificar, a sintaxe XTM para uma associação do tipo *aluno-escreve-artigo* (linha 3) entre *Giovani Librelotto* (linha 9), desempenhando o papel *aluno* (linha 7), e *Metamorphosis* (linha 15), desempenhando o papel *artigo* (linha 13), pode ser assim representada:

```

1 <association id="giovani-escreve-metamorphosis">
2   <instanceOf>
3     <topicRef xlink:href="#aluno-escreve-artigo"/>
4   </instanceOf>
5   <member>
6     <roleSpec>
7       <topicRef xlink:href="#aluno"/>
8     </roleSpec>
9     <topicRef xlink:href="#gr1"/>
10  </member>
11  <member>
12    <roleSpec>
13      <topicRef xlink:href="#artigo"/>

```

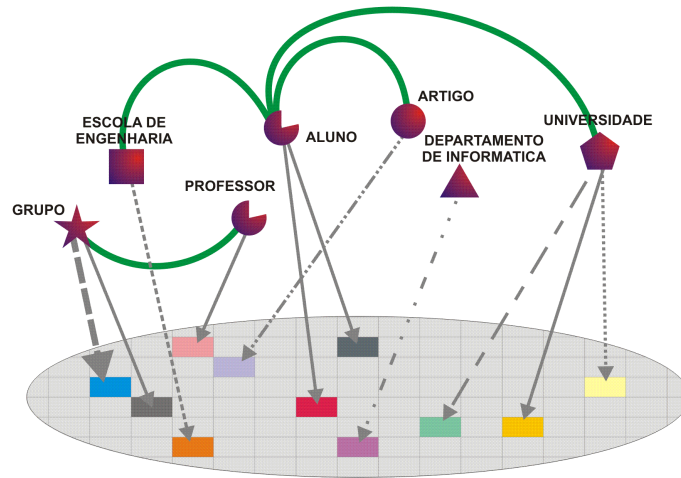


Figura 3.7: Associações de Tópicos

```

14     </roleSpec>
15     <topicRef xlink:href="#metamorphosis"/>
16 </member>
17 </association>

```

Ao contrário dos tópicos, que possuem obrigatoriamente um identificador (atributo @id, linha 1), as associações não necessariamente devem ter seu identificador definido.

As associações entre tópicos podem ser agrupadas de acordo com o seu tipo, da mesma forma como os tópicos e as ocorrências. Também como nos outros casos, o tipo de associação é um tópico. Assim, a definição de um tipo de associação é feita, na linguagem XTM, no elemento `<topic>`; como tal, informa o nome da associação, além dos nomes dos papéis de actuação dos seus membros. Cada nome encontra-se em um elemento `<baseName>`, dentro de um contexto particular (`<scope>`), conforme apresentado a seguir:

```

1 <topic id="aluno-escreve-artigo">
2   <baseName>
3     <baseNameString>Aluno escreve artigos</baseNameString>
4   </baseName>
5   <baseName>
6     <scope>
7       <topicRef xlink:href="#aluno"/>
8     </scope>
9     <baseNameString>escreve o artigo</baseNameString>
10  </baseName>
11  <baseName>
12    <scope>
13      <topicRef xlink:href="#artigo"/>
14    </scope>
15    <baseNameString>é escrito por</baseNameString>
16  </baseName>
17 </topic>

```

Na definição formal do tópico que corresponde ao tipo de associação acima *aluno-escreve-artigo* (linha 1), introduz-se na linha 3 o seu nome geral; na linha 9, o nome do papel que

desempenhará o membro do tipo de tópico *aluno* (linha 7); por fim, na linha 15, o nome do papel que desempenhará o membro do tipo de tópico *artigo* (linha 13).

Assim, a leitura da associação identificada por *giovani-escreve-metamorphosis*, de acordo com seu tipo definido no tópico *aluno-escreve-artigo* é realizada da seguinte forma:

- Aluno Giovanni escreve o artigo Metamorphosis.
- Artigo Metamorphosis é escrito pelo aluno Giovanni.

### 3.1.3 Nomes de Tópicos (*Topic Names*)

Como foi dito, o identificador do tópico é a sua designação única, que permite o individualizá-lo, isto é, identificá-lo unicamente dentro do conjunto de tópicos de um topic map. É, portanto, a designação utilizada pelas aplicações, ou programas de computador.

O **nome** (*basename*), que podem ser um ou mais em cada tópico, permite associar ao tópico uma designação que ajude os humanos a entender o seu significado. Sem um nome, um tópico poderia somente ter um identificador, o que nem sempre é apropriado para a comunicação humana.

O **nome** é uma propriedade importante de um tópico. A capacidade de especificar mais que um nome de tópico pode ser usado para dois fins:

1. fornecer descrições alternativas, facilitando o entendimento de seu significado;
2. estabelecer designações a ser usadas para indicar o uso de nomes diferentes em contextos diferentes, como idioma, domínio, área geográfica, período histórico, etc.

Um exemplo para a definição de um **nome** para o tópico *artigo*, de acordo com a sintaxe XTM, pode ser o seguinte:

```
1 | <topic id="artigo">
2 |   <basename>
3 |     <baseNameString>Artigo</baseNameString>
4 |   </basename>
5 | </topic>
```

Como já referenciado, tópicos representam temas em computadores. Assim sendo, os **nomes de tópicos** podem ser vistos como legendas (descrições) referentes ao tema representado pelo tópico. Contudo, nomes são completamente ambíguos, pois diferentes pessoas podem relacionar o mesmo nome a diferentes temas.

Essa característica é importante quando dois ou mais topic maps são unidos em um só, pois se dois tópicos representam o mesmo tema, eles formarão o mesmo tópico no topic map resultante. Para evitar tais ambiguidades, a norma **Topic Maps** oferece duas opções para qualificar os **nomes** de um tópico, as quais serão apresentadas nas próximas subsecções: **nomes variantes** (subsecção 3.1.4) e **contexto** (subsecção 3.1.5).

### 3.1.4 Nomes Variantes

**Nomes variantes** (*variant names*) são formas alternativas de nomes para visualização ou objectivos específicos, os quais dependem dos propósitos do topic map em questão.

O propósito de um **nome variante** é definido pelos seus parâmetros. Diferentemente do contexto (*scope*) – o qual define o âmbito da aplicabilidade das características dos tópicos – os parâmetros definem a forma como será usado o nome variante. Um **nome variante** pode ter sub-variantes suportando uso em cascata (por exemplo, diferentes nomes de apresentação para diferentes dispositivos de saída).

Um nome de apresentação pode ser um texto literal ou um recurso referenciado. O recurso referenciado pode conter qualquer tipo de dados, que pode ser usado para exibir o nome de um tópico (por exemplo, ícones, arquivo de áudio, etc.).

Para exemplificar, os seguintes **nomes variantes** podem ser usados para o tópico *República Federativa do Brasil* (ou simplesmente *Brasil*), para sua visualização em um dispositivo WAP<sup>4</sup>:

**Nome do Tópico:** República Federativa do Brasil

**Texto para WAP:** Brasil

**Imagem para WAP:** [http://www.deltatranslator.com/bandeira\\_do\\_brasil.gif](http://www.deltatranslator.com/bandeira_do_brasil.gif)

**Som para WAP:** [http://www.laurapoesias.com/poetas/hino\\_nacional\\_brasileiro.mid](http://www.laurapoesias.com/poetas/hino_nacional_brasileiro.mid)

**Som para MP3 Player:** <http://www.brasilrepublica.hpg.com.br/hinonacional.mp3>

Aplicando o exemplo acima, os nomes variantes dos tipos *imagem para WAP* e *Som para MP3 Player* são representados na sintaxe XTM da seguinte maneira:

```

1 <topic id="brasil">
2   <baseName>
3     <baseNameString>República Federativa do Brasil</baseNameString>
4     <variant>
5       <parameters>
6         <topicRef xlink:href="#ImagemWAP"/>
7       </parameters>
8       <variantName>
9         <resourceRef xlink:href="http://www.deltatranslator.com/bandeira_do_brasil.gif"/>
10      </variantName>
11    </variant>
12    <variant>
13      <parameters>
14        <topicRef xlink:href="#MP3Player"/>
15      </parameters>
16      <variantName>
17        <resourceRef xlink:href="http://www.brasilrepublica.hpg.com.br/hinonacional.mp3"/>

```

<sup>4</sup>O protocolo WAP (*Wireless Application Protocol*) (Niskanen, 2000) é uma norma para o serviço de informação em terminais sem fio (*wireless*), tal como telefones móveis digitais.

```

18 |         </variantName>
19 |     </variant>
20 | </baseName>
21 | ...
22 | </topic>

```

Este exemplo representa dois nomes variantes para o tópico *Brasil*, cujo nome base é *República Federativa do Brasil*. O primeiro nome variante é uma imagem que faz referência ao recurso descrito na linha 9, a qual será visualizada por um dispositivo WAP; o segundo nome variante é um recurso a ser acedido por um dispositivo MP3 (*Moving Picture Experts Group Layer-3 Audio*) (Hacker, 2000), descrito na linha 17.

### 3.1.5 Contexto (*Scope*)

A norma Topic Maps além de associar um identificador único a cada tópico, permite dizer, basicamente, três coisas sobre qualquer tópico: quais os seus nomes, quais são suas ocorrências e em que associações participa. Estes três tipos de afirmações são conhecidos como *características de tópico*.

Uma faceta bem interessante de Topic Maps é o facto de todas as características de tópico possam estar associadas a um **contexto** específico, que pode ou não estar explícito. Declarações semelhantes podem causar ambiguidade; nesse caso o conceito de **contexto** é usado para eliminar tais problemas. A declaração de **contextos** é realizada com a etiqueta `<scope>`.

Os **contextos** podem ser usados para definir diferentes perspectivas em um mesmo conjunto de informação. Algumas aplicações de **contexto** incluem:

**Idiomas:** pode-se usar contexto para diferenciar nomes e ocorrências em *Português* de nomes e ocorrências em *Italiano* ou *Espanhol*, por exemplo;

**Audiência:** o contexto pode ser usado para diferenciar recursos para utilizadores *iniciantes* de recursos para utilizadores *intermediários* ou *avançados*, permitindo assim que diferentes conjuntos de informação sejam apresentados para utilizadores de diferentes níveis;

**Tempo:** algumas propriedades sofrem mudanças conforme o tempo passa. Por exemplo, muitas cidades tiveram nomes distintos ao longo do tempo;

**Autoria:** o contexto pode ser utilizado para determinar quem disse o que; nomes e ocorrências em algum contexto específico podem ser usadas pelos leitores do topic map para determinar em quais partes da informação podem acreditar.

Devido ao facto de que **contexto** é um conceito poderoso e genérico, Topic Maps não possuem um conjunto pré-definido de contextos. Todavia, a norma permite que autores definam-os usando tópicos como construtores.

A Figura 3.8 ilustra o uso de **contexto** para diferenciar o idioma de alguns nomes de tópicos referentes ao tema *artigo*: em *Italiano*, usa-se a palavra *Articulo*, enquanto que em *Inglês*, usa-se *Paper*.

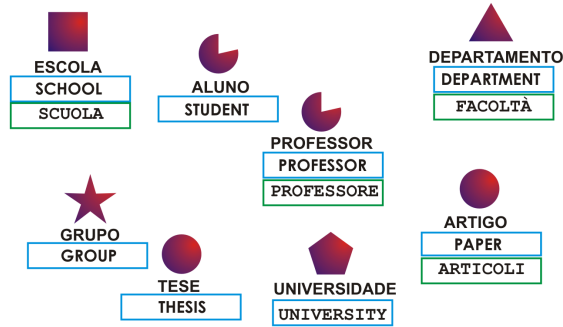


Figura 3.8: Nomes de tópicos em diversos contextos

A título de exemplo, considera-se o tópico *artigo* representado em XTM da maneira abaixo descrita:

```

1 <topic id="artigo">
2   <baseName>
3     <baseNameString>Artigo</baseNameString>
4   </baseName>
5   <baseName>
6     <scope>
7       <topicRef xlink:href="#EN"/>
8     </scope>
9     <baseNameString>Paper</baseNameString>
10  </baseName>
11  <baseName>
12    <scope>
13      <topicRef xlink:href="#IT"/>
14    </scope>
15    <baseNameString>Articulo</baseNameString>
16  </baseName>
17 </topic>

```

O código XTM acima apresenta o tópico *artigo* contendo três nomes: um nome sem contexto – *Artigo* (linhas 2 a 4); um nome que somente é válido no contexto *EN* – *Paper* (linhas 5 a 10); e um nome que somente é válido no contexto *IT* – *Articulo* (linhas 11 a 16).

O uso da etiqueta `<scope>` faz com que os nomes sejam activos ou não-activos em um determinado momento. Por exemplo, ao estar situado no contexto *EN*, os nomes contidos nos demais contextos não são visualizados; o mesmo vale para todas as características de tópicos inseridas em algum contexto num topic map. Por sua vez, os nomes sem contexto (*unconstrained scope*) são válidos em todos os contextos.

Por exemplo, o nome *Artigo* será válido tanto no contexto de partida (*unconstrained scope*) como nos contextos *EN* e *IT*; por sua vez, o nome *Paper* só será válido no contexto *EN* e em nenhum outro mais.



O **contexto**, sendo bem aplicado, permite muito mais que eliminar ambiguidades; pode também servir como ajuda na navegação, permitindo alterar dinamicamente a visualização de um topic map baseado num determinado perfil de utilizador, no qual o contexto pode usado por sistemas de filtragem que mostrarão diferentes propriedades dos tópicos, conforme o domínio que o topic map está descrevendo.

## 3.2 Fusão de Topic Maps

Fusão (*merge*) é o processo de unir topic maps, fundindo adequadamente os tópicos de cada um. Em XTM, o processo de fusão de dois topic maps é definido no elemento `<mergeMap>`.

Quando há uma fusão entre dois topic maps, todos os tópicos com o mesmo identificador de tema são fundidos, tornando-se um único tópico no topic map resultante. As características do tópico resultante serão todas as características dos tópicos originais, implicando que todos os nomes, ocorrências e associações duplicados são removidos.

Para exemplificar, a Figura 3.9 mostra dois topic maps antes da união, enquanto que a Figura 3.10 apresenta o topic map resultante.

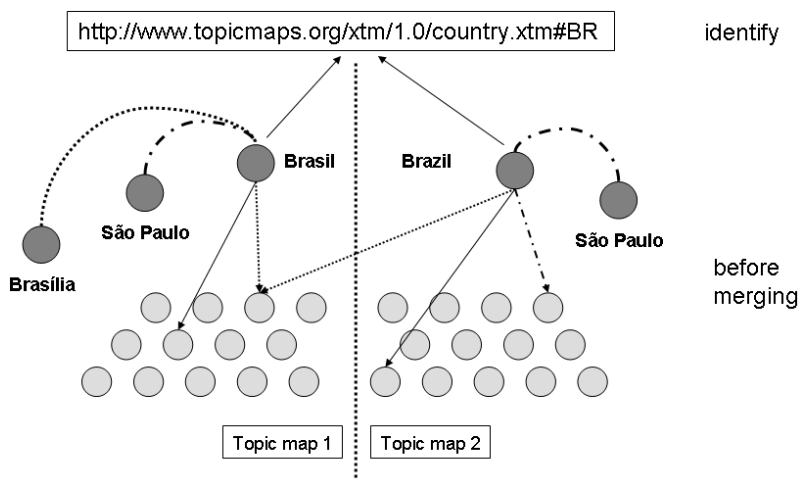


Figura 3.9: Dois topic maps antes da união. Fonte: The Topic Maps Handbook (2003).

As regras para a fusão de Topic Maps são:

1. de acordo com *Topic Naming Constraint* (TNC) (Mason and Desautels, 2002), dois tópicos são fundidos se possuem um nome idêntico num mesmo contexto;
2. de acordo com as *regras de fusão baseadas em temas* (Park and Hunting, 2003), os tópicos são fundidos se tiverem a mesma identidade de tema, ou seja, se os sub-elementos `<subjectIdentify>` apontam para o mesmo recurso;

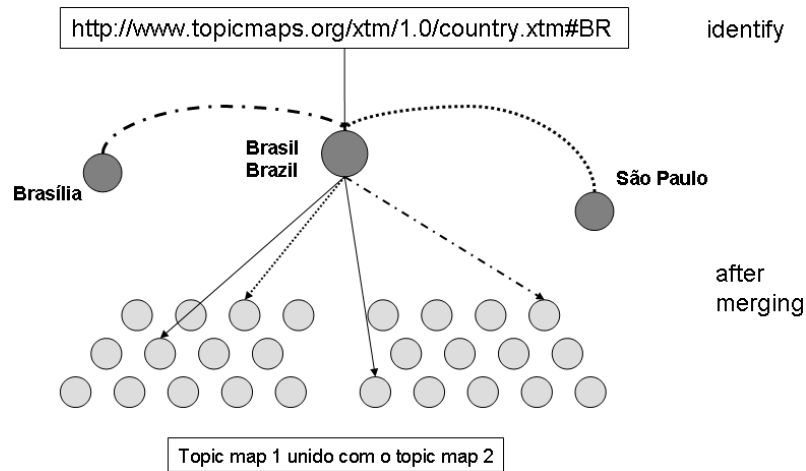


Figura 3.10: Dois topic maps após da união. Fonte: The Topic Maps Handbook (2003).

As regras de fusão baseadas em temas tem o objectivo de encorajar os autores de topic maps, que desejam partilhar ontologias, a referir estes temas publicados na Web.

Pois um desafio de **Topic Maps** é justamente representar um tema do mundo real (abstracto ou real) em forma de tópicos. Isto porque ao nomear um tópico como *Giovani Rubert Librelotto* não necessariamente tem-se a certeza que outros sistemas computacionais (ou outras pessoas) saberão que está a se referir exactamente ao brasileiro, aluno de doutoramento do Departamento de Informática da Universidade do Minho e orientando do professor Pedro Rangel Henriques.

Podem existir também casos onde um mesmo tema seja representado por mais de um tópico. Isso ocorre principalmente após a fusão de topic maps. Nessa situação, é necessário possuir formas de estabelecer a identidade entre tópicos aparentemente discrepantes.

Por exemplo, se ocorrer o caso de fundir topic maps sobre a Igreja Católica que estão em diferentes idiomas (inglês, português e italiano), há uma necessidade de afirmar que os tópicos *John Paul II*, *João Paulo II* e *Giovanni Paolo II* referem-se ao mesmo tema. Se houver um recurso que identifique este tema, este recurso é definido como **tema publicado** (*Published Subjects*) (Pepper, 2003; OASIS, 2003b; OASIS, 2003a).

Os **temas publicados** são necessários para normalizar e permitir a reutilização de temas em vários topic maps, facilitando a fusão dos mesmos. O uso dos **temas publicados** depende de certas condições, tais como:

**Publicidade e Visibilidade:** o autor do topic map tem que ter conhecimento da existência do tema desejado;

**Concordância:** o autor do topic map tem que concordar que o tema descreve realmente o conceito que o autor tem em mente.

**Fiabilidade e Estabilidade:** o autor do topic map necessita ter a garantia que o tema é acessível e imutável, pelo menos durante o período da existência do topic map.

Os repositórios (websites) onde são colocados e mantidos os temas publicados são chamados de *Indicadores de Temas Publicados – Published Subject Indicator* (PSI) (Nishikawa, 2003).

Um **tema publicado** é um tema que teve disponibilizado um indicador para seu uso público, o qual está acessível através de uma URI. Um PSI é então um recurso que tenha sido publicado para fornecer facilidades de fusão e intercâmbio de topic maps.

Um bom exemplo de PSI para topic maps é <http://www.topicmaps.org>, o qual é estável e os utilizadores encontram uma clara definição de topic maps, uma discussão sobre sua história, o vocabulário para topic maps e a especificação XTM, tudo isso validado por especialistas da área.

### 3.3 Identificador e Indicador de Tema: definição da identidade de um tema

Um **identificador de tema** (*subject identifier*) é uma URI que representa um tema não-endereçável. Como exemplo de **identificadores de temas**, apresentam-se:

**Tema – Brasil:** <http://www.topicmaps.org/xtm/1.0/country.xtm#BR> – conforme definido pela ISO 3166 (ISO, 1997);

**Tema – Idioma Português:** <http://www.topicmaps.org/xtm/1.0/language.xtm#PT> – conforme definido pela ISO 639 (ISO, 1998).

**Tema – Idioma Inglês:** <http://www.topicmaps.org/xtm/1.0/language.xtm#EN> – conforme definido pela ISO 639 (ISO, 1998).

**Tema – Idioma Italiano:** <http://www.topicmaps.org/xtm/1.0/language.xtm#IT> – conforme definido pela ISO 639 (ISO, 1998).

Se o **identificador de tema** (*subject identifier*) é uma URI que aponta para um recurso, o recurso em questão é chamado **indicador de tema** (*subject indicator*). O recurso deve conter uma documentação legível por humanos que descreve o tema não-endereçável, como por exemplo, um documento, um vídeo, um áudio, um topic map, etc.

Na Figura 3.11, apresenta-se um exemplo do tópico *Brasil*. Por ser um tema não-endereçável – não pode ser acedido electronicamente – define-se o **identificador de tema** <http://www.topicmaps.org/xtm/1.0/country.xtm#BR> que estabelece a identidade deste tópico. Assim, no momento de uma fusão deste topic map com outro que tenha o tópico *Brasil* em seu domínio, o facto de ambos usarem o mesmo **identificador de tema** será determinante para sua união.

Portanto, um *indicador de tema* descreve um tema com o propósito deste ser amplamente usado pelos autores de **Topic Maps**. Consequentemente, é muito importante que os autores escrevam os indicadores de temas de uma maneira clara e concisa, salientando exactamente que tema está sendo abordado.

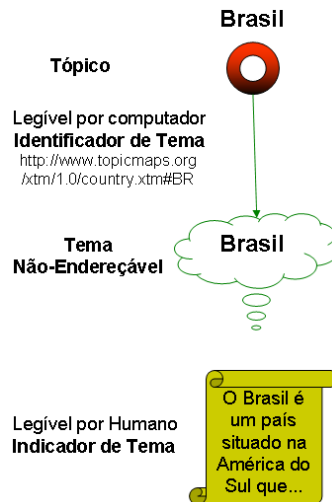


Figura 3.11: Temas não-endereçáveis. Fonte: The Topic Maps Handbook (2003).

Na prática, um **indicador de tema** é um recurso que tem a intenção de fornecer ao autor do topic map uma indicação positiva e não-ambígua da **identidade de um tema** (Park and Hunting, 2003).

A **identidade de tema** (*subject identity*) conecta o tópico com o tema que o tópico representa. O exemplo da Figura 3.12 mostra dois topic maps que possuem tópicos que têm o **identificador de tema** apontando para o **indicador de tema** <http://www.topicmaps.org/xtm/1.0/country.xtm#BR>. Isto indica que ambos se referem ao mesmo tema; portanto, podem ser fundidos em um único tópico, em uma eventual fusão.

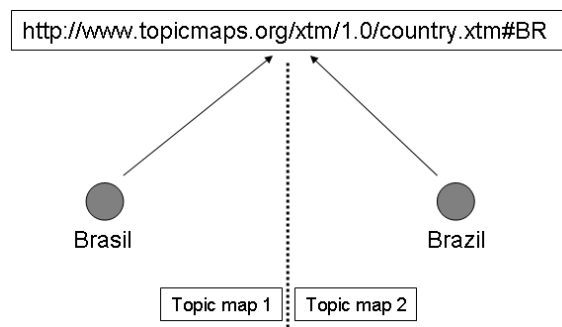


Figura 3.12: Identidade de dois topic maps. Fonte: The Topic Maps Handbook (2003).

A **identidade de tema** pode ser uma referência para:

- um recurso, o qual é um tema endereçável electronicamente;
- um identificador de tema, o qual identifica um tema não-endereçável electronicamente;
- um outro tópico, o qual compartilha o mesmo tema com o tópico corrente.

Abaixo, apresenta-se a sintaxe XTM para cada um dos items acima citados:

```
1 <topic id="brasil-gov-br">
2   <subjectIdentity>
3     <resourceRef xlink:href="http://www.brasil.gov.br/">
4   </subjectIdentity>
5 </topic>
6 <topic id="brasil">
7   <subjectIdentity>
8     <subjectIndicatorRef xlink:href="http://www.topicmaps.org/xtm/1.0/country.xtm#BR"/>
9   </subjectIdentity>
10 </topic>
11 <topic id="brazil">
12   <subjectIdentity>
13     <topicRef xlink:href="#brasil"/>
14   </subjectIdentity>
15 </topic>
```

De acordo com o código XTM acima, três tópicos são definidos:

1. Tópico reificando o tema *Governo Brasileiro*; a identidade deste tema é o recurso especificado na linha 3;
2. Tópico reificando o tema *Brasil*; a identidade deste tema é uma referência ao indicador de tema apresentado na linha 8;
3. Tópico reificando o tema *Brasil*; como outro tópico faz uma referência ao indicador deste tema, basta fazer uma referência a este tópico, como realizado na linha 13.

Com isso, pode-se afirmar que a identidade de tema é definida na etiqueta `<subjectIdentity>` da sintaxe XTM.

## 3.4 Linguagens concretas para Topic Maps

A notação, ou linguagem, mais usada para a escrita de Topic Maps é XTM, a qual foi apresentada na Subsecção 3.1.1. Essa secção irá abordar das restantes notações.

### 3.4.1 A sintaxe HyTM

A sintaxe HyTM (*HyTime Topic Maps*) (Bryan, 2003) foi definida na norma Topic Maps original ISO 13250. É um dialecto SGML baseado em HyTime (Goldfarb et al., 1997) (então a origem do nome).

HyTM é menos poderoso (e extenso) que XTM. Actualmente, HyTM é utilizado em poucos projectos, visto que XTM foi adoptado como norma pela comunidade académica.

A sintaxe HyTM (Newcomb, Biezunski, and Bryan, 2003) foi definida para a criação de topic maps expressados em SGML ou XML, de acordo com a estrutura definida por um DTD. Um exemplo de topic map segundo a sintaxe HyTM é abaixo apresentado:

```

1 <topicmap id="tm-exemplo">
2   <topic id="giovani" types="pessoa">
3     <topname>
4       <basename>Giovani Rubert Librelotto</basename>
5     </topname>
6     <occurs type="curriculum" href="http://www.di.uminho.pt/~grl/lattes/">
7   </topic>
8   <topic id="prh" types="pessoa">
9     <topname>
10      <basename>Pedro Rangel Henriques</basename>
11    </topname>
12    <occurs type="email" href="mailto:prh@di.uminho.pt">
13  </topic>
14  ...
15 </topicmap>

```

A notação acima define dois tópicos do tipo *pessoa*: o primeiro (linhas 2-7) possui o identificador *giovani*, o nome *Giovani Rubert Librelotto* e uma ocorrência do tipo *curriculum*; o segundo (linhas 8-13) possui o identificador *prh*, o nome *Pedro Rangel Henriques* e uma ocorrência do tipo *email*.

Abaixo demonstra-se a definição de uma associação entre estes tópicos, a qual especifica uma orientação: Pedro Rangel Henriques é orientador de Giovani Rubert Librelotto, enquanto que Giovani é orientando de Pedro Henriques.

```

1 <assoc type="orientar">
2   <assocrl type="orientador" href="prh"/>
3   <assocrl type="orientando" href="giovani"/>
4 </assoc>

```

Em termos de comparação, XTM difere de HyTM em alguns detalhes, listados a seguir:

- Usa XML, enquanto HyTM é baseado em SGML;
- Define um simples DTD para sua validação;
- Elimina o elemento *facet* de HyTM, pois sua funcionalidade pode ser realizada pelo elemento *association*;
- Generaliza os elementos *sortName* e *dispName* de HyTM em *variant* enquanto preserva a semântica dos elementos de HyTM através de *Published Subject Indicators* (PSI) no elemento *parameter*, sub-elemento de *variant*;
- Introduce a distinção entre *subject-indicating* e *subject-constituting*<sup>5</sup>;
- Usa a sintaxe URI (*Uniform Resource Identifiers*) (IETF, 1998) XLink, enquanto que HyTM permite arbitrariedade ao endereçar esquemas;
- Usa um estilo mais claro de nomes de *tags* (por exemplo, *association* ao invés de *assoc*);

<sup>5</sup>Para uma discussão detalhada, ver os capítulos 6 e 7 de (Park and Hunting, 2003)

- Usa elemento ao invés de atributos enquanto possível, para a representação dos conceitos da norma Topic Maps.

Devido a estes factores, tanto esta dissertação como as ferramentas desenvolvidas em seu âmbito, adoptarão a sintaxe XTM.

### 3.4.2 A sintaxe AsTMa=

*AsTMa=* (*Asymptotic Topic Maps*) (Barta, 2004) é parte da família de linguagens *AsTMa* (Barta, 2003b) a qual foi projectada para facilitar a escrita, restrições e interrogações sobre topic maps.

A especificação abaixo apresenta um exemplo de um simples topic map declarado em *AsTMa=*:

```

1 | giovani bn : Giovanni Rubert Librelotto oc : http://www.di.uminho.pt/~grl
3 | prh bn : Pedro Rangel Henriques
5 | (orientar) orientador : prh orientando : giovani

```

O primeiro bloco no código acima (linhas 1–3) declara um tópico identificado por *giovani* que possuirá um nome *Giovanni Rubert Librelotto* e uma ocorrência apontará para um recurso de informação, no caso, o sítio internet <http://www.di.uminho.pt/~grl>. O segundo bloco (linhas 5–6) declara um tópico com o identificador *prh* com o nome *Pedro Rangel Henriques*.

O bloco final apresenta uma associação entre estes tópicos, definida pelo identificador *orientar*. O membro *prh* está desempenhando o papel de actuação *orientador*, enquanto que o membro *giovani* está desempenhando o papel de actuação *orientando*. Como é definido pela norma que qualquer tipo de associação e papel de actuação são tópicos, então também *orientar*, *orientador* e *orientando* terão que ser previamente definidos como tópicos.

Uma característica importante de Topic Maps é a união: dois ou mais topic maps podem ser unidos em um único topic map. As regras que definem esta união estão definidas na norma ISO 13250 (Biezunsky, Bryan, and Newcomb, 1999) e são respeitadas em XTM (Pepper and Moore, 2001a). Actualmente, ao contrário de XTM, *AsTMa=* não fornece nenhum mecanismo para união de topic maps.

### 3.4.3 A sintaxe LTM

A notação LTM (*Linear Topic Map*) (Garshol, 2002c) é uma sintaxe simples para Topic Maps. Tal como a sintaxe XTM, ela representa os construtores de um topic map como texto, mas em um formato mais simplificado e compacto. Esta notação pode ser escrita em qualquer editor de texto e processada por processadores de topic maps que suportam esse formato, ou pode ser convertida para o formato XTM.

A justificação para definir esta nova sintaxe LTM foi a necessidade de suprir as dificuldades de XTM, devido ao seu custo de produção e a dificuldade de leitura e escrita por humanos. Então se fazia necessário possuir uma sintaxe simples que pudesse ser usada para uma rápida criação de pequenos topic maps para demonstrações e propósitos pessoais. Esta sintaxe pode ser útil enquanto bons editores de topic maps não estão disponíveis.

Esta notação desenvolvida pela Ontopia<sup>6</sup>, oferece uma notação fácil e conveniente para gerir os topic maps. Contudo, ressalta-se que o único formato normalizado para intercâmbio de topic maps continua sendo o formato XTM 1.0.

A definição de um tópico é realizada escrevendo um identificador do tópico entre parêntesis rectos. Para associar nomes ao tópico, pode-se fazer como se mostra no exemplo abaixo, onde dois nomes são declarados:

```
1 | [giovani = "Giovani Rubert Librelotto";
2 |     "Rubert Librelotto, Giovani"]
```

Os tópicos também podem possuir tipos. No exemplo abaixo, define-se que *giovani* é uma instância de *pessoa*:

```
1 | [giovani : pessoa = "Giovani Rubert Librelotto";
2 |     "Rubert Librelotto, Giovani"]
```

Obviamente, o tópico identificado por *pessoa* está definido em algum lugar da especificação LTM.

LTM também suporta a definição de ocorrências. Isto é realizado usando a notação abaixo apresentada, onde as ocorrências estão entre chavetas. As ocorrências em LTM possuem três partes: o identificador do tópico que contém a ocorrência, o identificador do tipo da ocorrência e o localizador da ocorrência, entre aspas.

```
1 | {giovani, curriculum, "http://www.di.uminho.pt/~grl/lattes/"}
```

Também é possível definir dados como ocorrências de tópicos, como se mostra abaixo:

```
1 | {giovani, idade, "27"}
```

Para a definição de associações, a sintaxe LTM também é simples. No exemplo abaixo, definiu-se a associação entre Giovani e Pedro Henriques, onde ambos estão associados a um papel de actuação em associação.

```
1 | orientar(giovani : orientando, prh : orientador)
```

Após o estudo exaustivo de LTM, ao efectuar-se uma comparação entre os construtores de Topic Maps definidos na norma ISO 13250 e em XTM, percebe-se que LTM não suporta:

---

<sup>6</sup><http://www.ontopia.net/>



- os nomes, ou alternativos;
- as facetas (encontradas apenas em HyTM, as quais estão obsoletas) – por isso não são suportadas por LTM;
- reificação de nomes, ocorrências, associações e papéis de actuação em associação; segundo os autores, estas facilidades podem ser suportadas em uma versão futura.

### 3.5 Justificativa da escolha de Topic Maps em vez de RDF

O foco desta secção é comparar as duas tecnologias, e usar esta comparação para justificar o porquê da escolha de **Topic Maps** como representação da estrutura do conhecimento nesta tese. Esta secção se inicia com uma visão histórica das normas e se completa com um comparativo entre ambas.

#### 3.5.1 Diferenças históricas

A norma **Topic Maps** surgiu no princípio da década de 90 associada aos trabalhos na área de gestão de índices para documentação e foi usada extensivamente por diversos anos antes de ser adoptado pela ISO (ISO, 2005), em 1996. Deste trabalho, resultou na publicação a norma **Topic Maps**, como ISO 13250, no princípio do ano 2000 (Biezunsky, Bryan, and Newcomb, 1999), e posteriormente, foi XTM 1.0 (Pepper and Moore, 2001a), em 2001 (Pepper, 1999).

RDF, por outro lado, origina-se em um trabalho feito por R.V. Guha, na Apple, no contexto do MCF (*the Meta Content Framework*), o qual tornou-se uma aplicação XML e foi adoptado pela W3C (W3C, 2005). A W3C transformou-a no que hoje conhecemos como RDF, o qual foi publicado como uma Recomendação W3C, em 1999.

Até 1999, as duas comunidades (ISO e W3C) não tinham conhecimento do trabalho uma da outra; quando ficaram sabendo, já era tarde para uma integração. A recomendação RDF já havia sido publicada, e **Topic Maps** havia sido submetida como norma ISO não muito tempo depois. As duas comunidades tentaram de algumas formas fundir as tecnologias, porém as especificações já estavam publicadas e isso iria implicar em modificar conceitos que já eram amplamente usadas e estavam bem compreendidos por cada uma das comunidades.

Eis a actual posição: tem-se duas comunidades técnicas, as quais possuem sua própria família de normas que estão fortemente estabelecidas. Uma fusão parece ser politicamente impossível, e como poderá ser visto nessa secção, há também argumentos técnicos para isto. Mantendo essa tendência, RDF e **Topic Maps** são vistos como tecnologias que competem entre si (Pepper, 2002).

### 3.5.2 Diferenças à primeira vista

Ambas as normas foram projectadas para diferentes objectivos. **Topic Maps** foi desenvolvido para suportar um alto nível de indexação de conjuntos de recursos de informação para tornar fácil a recuperação dessa informação e a extracção de conhecimento. A linguagem **RDF**, por outro lado, foi projectado para suportar a visão semântica da Web (*Semantic Web*), fornecendo metadados estruturados sobre os recursos e uma fundamentação para inferência lógica.

As diferenças em perspectiva entre essas duas tecnologias, tornam difícil uma fusão entre ambas, pelo facto de que as comunidades não tem os mesmos objectivos. As diferenças também fornecem um argumento para a separação entre ambas: diferentes ferramentas para objectivos distintos.

### 3.5.3 Entendendo as normas

Ao olhar para **Topic Maps** ou **RDF**, percebe-se uma quantidade enorme de acrónimos relacionados, o que pode tornar difícil o entendimento de como o nodo X relaciona-se com o nodo Y nesta “sopa de letras”. Na Figura 3.13 apresenta-se um diagrama que mostra as principais siglas relacionadas com as normas **Topic Maps** e **RDF**.



Figura 3.13: As famílias das duas normas. Fonte: Living with topic maps and RDF (2003).

A Figura 3.13 divide as duas famílias em três áreas: sintaxes (linguagens concretas usáveis para a descrição dos modelos); modelo lógico de dados; e restrições semânticas (*constraints*). Tanto **RDF** como **Topic Maps** tem um considerável número de linguagens; algumas padronizadas, outras não. Nesta tese, não serão abordados em detalhe estes formatos (excluindo-se **XTM**, obviamente), pois a interoperabilidade neste nível não é útil nesta secção, quando o que realmente importa é o alto lógico de representação.

Quando se refere a norma **Topic Maps**, refere-se sobre o modelo abstracto de mapas de tópicos (topic maps), o qual consiste de tópicos, associações, ocorrências, e outros. Da mesma forma com **RDF**, o qual consiste de nodos e declarações. Este é o nível (nível de modelo) ao qual se necessita comparar ambos para entender o relacionamento entre eles.

**Topic Maps** e **RDF** são famílias de normas que incluem mais do que apenas sintaxe e modelos de dados. Há também linguagens de restrições – e em algum ponto no futuro, haverá também linguagens de *query*.

Em termos de especificação de restrições, TMCL está para a norma **Topic Maps**, da como RDF Schema/OWL está para RDF.

Como anteriormente mencionado, **Topic Maps** e RDF são estruturalmente diferentes; consequentemente, significa que as restrições devem ser especificadas diferentemente nestas duas linguagens. Por exemplo, TMCL deve permitir restringir o contexto das características dos tópicos, assim como os papéis de actuação em associações, o que não existe em RDF.

OWL e TMCL não estão finalizados até o momento, o que torna possível evitar incompatibilidades no projecto. Infelizmente, isto torna difícil a criação de mapeamento entre ambas. Apesar disso, Lars Marius Garshol propõe duas abordagens diferentes para este fim (Garshol, 2003):

**Converter ao nível do Modelo de Dados:** validar um topic maps usando um RDF Schema/OWL e mapeando este vocabulário de RDF para **Topic Maps** (e vice-versa);

**Converter ao nível da Linguagem de Restrição:** converter directamente a RDF Schema/OWL para TCML, usando o mapeamento de RDF para **Topic Maps** para conduzir a conversão (novamente, vice-versa).

OIL e DAML não foram mencionadas na Figura 3.13 pela simples razão de que, como se disse atrás, eles foram sucedidos por OWL.

### 3.5.4 Comparando as normas

Esta secção tem por objectivo demonstrar potenciais similaridades e diferenças entre RDF e **Topic Maps**. As características aqui apresentadas foram importantes factores na escolha de **Topic Maps** para a representação de ontologias.

RDF e **Topic Maps** possuem certas características similares. Ambos foram desenvolvidos como técnicas de representação do conhecimento para a gestão da informação. Ambas definem modelos abstractos e possuem linguagens concretas baseadas em XML; ambas também possuem modelos que são simples e elegantes em um nível, mas extremamente poderosos em outro: os conceitos, em **Topic Maps**, são representadas como tópicos; enquanto que em RDF, elas são representadas como recursos.

Contudo, há uma série de diferenças entre RDF e **Topic Maps**. Uma das principais é que, em **Topic Maps**, a representação do conhecimento é destinado à gestão da informação por humanos. Essa noção pode ser exemplificada ao comparar **Topic Maps** com um índice de um livro, onde as informações estão todas associadas de acordo com o seu significado no contexto onde ela está inserida. RDF representa o conhecimento destinado à gestão de informação por máquinas (computadores).

**Topic Maps** e RDF também possuem pontos-de-vista distintos, pois enquanto que RDF é centrado em recursos (*resource-centric*), **Topic Maps** é centrado em temas (*subject-centric*). Em RDF, inicia-se com os recursos de informação para posteriormente anexar estruturas de metadados a eles; em **Topic Maps** foca-se no tema a que informação se está referindo.

Assim, recurso em RDF e tema em **Topic Maps** podem ser entendidos como sinónimos, desde que os recursos de informação sejam endereçáveis<sup>7</sup>.

Martin Lacher e Stefan Decker (Lacher and Decker, 2001) apresentam uma maneira para conseguir interoperabilidade entre **Topic Maps** e RDF, a qual habilita consultas em recursos de informação RDF e **Topic Maps**. Essas consultas são obtidas através da representação de grafos internos para **Topic Maps**.

Uma outra proposta de integração entre **Topic Maps** e RDF através de uma colecção de três ferramentas para a gestão de meta-informação em RDF e **Topic Maps**, chamada META, é apresentada em (Ciancarini et al., 2003). Esta proposta ressalta que muitos construtores naturais em RDF não tem estrutura correspondente em **Topic Maps**, e vice-versa. Isto indica que a conversão automática pode ser impraticável. Assim, META não consegue resolver o problema de que, em determinadas situações, é necessário estender RDF para obter o mesmo poder de expressividade de **Topic Maps**.

### Diferentes níveis semânticos

Em relação ao nível semântico, pode-se afirmar que RDF está num nível mais baixo que **Topic Maps**, pois em RDF os recursos possuem propriedades que têm valores (os quais podem ser outros recursos), e só; enquanto que em **Topic Maps**, os tópicos possuem características de vários tipos, como nomes, ocorrências e papéis de actuação em associações com outros tópicos. Portanto, a falta desta distinção semântica entre essas características é sentida em RDF. Aliado a isto, várias experiências mostram que os humanos concordam que este nível semântico é muito mais intuitivo (Pepper, 2002). Em adição, a diferença no nível semântico também interfere no facto de que um browser genérico (como o *Ontopia Omnigator* (Ontopia, 2002b) e o *Ulysses* (Librelotto, Ramalho, and Henriques, 2004c)) pode ser construído para **Topic Maps**, mas não para RDF.

### Mapeamento genérico RDF x TM e vice-versa

Para tornar os modelos mais próximos, é possível definir um mapeamento genérico entre ambos, nas duas direcções. Contudo, os resultados práticos obtidos não são muito bons, pois uma tripla RDF pode ser, em teoria, mapeada em até seis diferentes termos em **Topic Maps**; assim, sem o conhecimento da semântica de predicados, uma escolha adequada não pode ser feita. Do mesmo modo, as características dos tópicos podem ser mapeadas genericamente para triplas RDF, mas o alto nível semântico é perdido na ausência de um RDF Schema para **Topic Maps** (Garshol, 2002a); e mesmo com o esquema, os resultados são totalmente inadequados do ponto de vista do processamento de RDF.

---

<sup>7</sup>Temas endereçáveis e não-endereçáveis serão discutidos na Subsecção 3.3

### Assuntos endereçáveis e não-endereçáveis

Outra importante característica ausente em RDF são os indicadores de tema (*subject indicators*). O assunto de cada declaração em um modelo RDF é um recurso, indicado por uma URI. Já em **Topic Maps**, o assunto de cada declaração é um tópico, representando um assunto, o qual pode ser endereçável ou não-endereçável. Os assuntos endereçáveis são identificados por suas URI's (como em RDF); os assuntos não-endereçáveis são identificados pelas URI's de um ou mais indicadores de temas (*subject indicators*<sup>8</sup>).

### Relacionamentos

Um ponto onde se percebe claramente as diferenças entre RDF e **Topic Maps** é nas declarações. Por exemplo, *estuda(UMinho, Giovanni)* é diferente de *estuda(UMinho, Gustavo)*, da mesma forma que é diferente de *é-estudante(Giovanni, UMinho)*, o que pode acarretar em redundância. Em **Topic Maps**, não é possível declarar que *Giovanni estuda na UMinho* sem também definir que *UMinho fornece ensino ao Giovanni* – isto encontra-se na mesma associação. Essa expressividade é possível graças a noção de papéis de actuação em associação, o que torna claro o papel de actuação de cada participante em um relacionamento.

As relações em RDF são sempre binárias, correspondendo a uma construção *sujeito-verbo-objecto* em linguagem natural. Por exemplo, *Giovanni estuda na UMinho* ou *estuda(UMinho, Giovanni)*. Em **Topic Maps**, as associações podem ser n-árias. Uma associação pode ter qualquer número de papéis de actuação e pode, assim, expressar facilmente relacionamentos mais complexos. Por exemplo, *Giovanni estuda na UMinho estando inscrito num Doutorado em Informática* ou *estuda(UMinho, Giovanni, Doutorado em Informática)*; ou associações unárias, como *Giovanni estuda* ou *estuda(Giovanni)*.

### Identidade

Identidade é a forma de descobrir quando dois tópicos representam o mesmo tema, como isto é feito e que consequências esta descoberta traz. Este conceito é especialmente útil quanto permite-se definir identidades automaticamente, com dados de diferentes recursos de informação. RDF e **Topic Maps** suportam este conceito, porém em diferentes formas.

Em RDF, há três tipos de nodos (Garshol, 2003):

**Literais:** um texto (*string*);

**Nodos URI:** nodo que contém uma URI identificando o recurso representado pelo nodo. RDF assume que os nodos com o mesmo URI representam o mesmo recurso;

**Nodos em branco:** nodos que não possui um rótulo URI. Para estes nodos, há uma única maneira de descobrir quais recursos eles representam: analisando as declarações

---

<sup>8</sup>Os indicadores de temas serão discutidos na Subsecção 3.3.

feitas sobre eles. RDF não fornece uma forma normalizada para identificar nodos em branco que representam os mesmos recursos; isto se obtém apenas com vocabulários de alto-nível, tal como OWL.

Em Topic Maps, tópicos podem representar *nodos em branco* e *nodos URI*. As URIs podem ser anexadas aos tópicos para identificar seus temas, tornando os tópicos iguais a nodos URI em RDF. Se nenhuma URI é designada, o tópico é exactamente como um nodo em branco.

Em RDF, quando uma URI é utilizada para identificar um tema, causa-se uma ambiguidade. Por exemplo, a URI <http://www.uminho.pt/> pode identificar um tema. Agora, o que isto realmente identifica? O recurso de informação obtido através da URI, ou seja, o próprio website? Ou a universidade descrita pelo recurso de informação?

Topic Maps distinguem ambos os casos. Quando um tópico define uma URI como um identificador, ela pode ser considerada como:

**Endereço de tema (*subject address*):** o tema é o próprio recurso;

**Identificador de tema (*subject identifier*):** o tema é o que é descrito pelo recurso (Pepper and Schwab, 2003).

Em RDF, porém, não existe esta distinção; um nodo URI não contém maneira de determinar sob qual das duas formas ele deve ser interpretado. Os modelos RDF consistem de declarações e recursos, ignorando o facto de que os recursos não são parte do modelo RDF, e sim, representados por nodos RDF.

## Reificação

Uma reificação é o acto de criar um símbolo que representa uma tema.

Em Topic Maps, isto se faz ao criar um tópico e então fornecendo-o um identificador de tema (*subject identifier* – Subsecção 3.3) que aponta para o nome, ocorrência ou associação que deseja-se que o identifique. Após isto, o tópico pode ser usado como outro qualquer, pois o processador de Topic Maps irá detectar o que o tópico representa e agir consequentemente. A Figura 3.14 mostra, por exemplo, a reificação dos temas *Giovani Librelotto* e *Universidade do Minho*.

Em RDF, o mecanismo funciona de uma forma diferente. Um nodo em branco é criado e dado o tipo *rdf:Statement*. Um RDF Schema é usado para adicionar o tema, propriedades e objectos através de declarações. Quando atravessa-se de um nodo a outro, através das declarações de uma determinada propriedade, as declarações reificadas são percorridas em uma forma diferente das não-reificadas, o que torna complexo o trabalho com reificação.

Conforme a Figura 3.15, reificações em RDF modificam as declarações, enquanto que a reificação não modifica as características de um tópico. O caminho de *Giovani Librelotto* para a *Universidade do Minho* permanece o mesmo em um topic map, mas é modificado em um diagrama RDF.

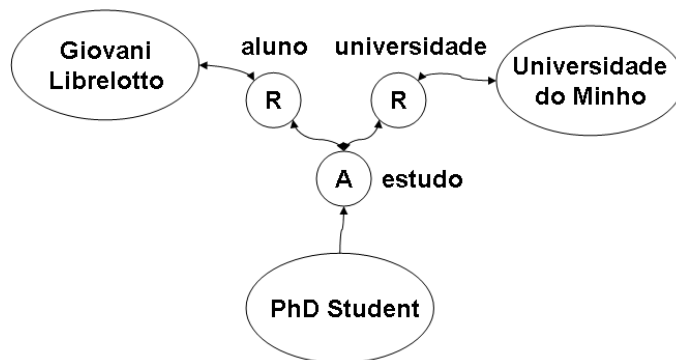


Figura 3.14: Reificando uma Associação. Fonte: Lars Marius Garshol (2003).

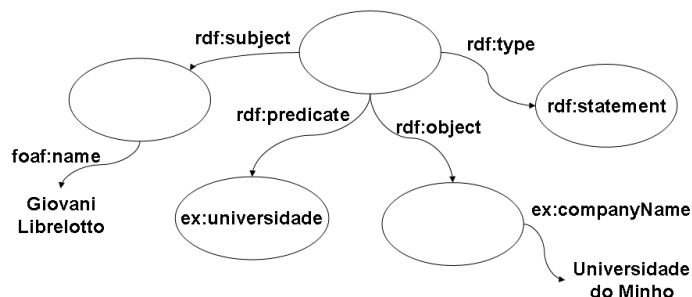


Figura 3.15: Reificando uma Declaração. Fonte: Lars Marius Garshol (2003).

Este problema é conhecido pela comunidade RDF, porém não há uma solução proposta conhecida até o momento (Garshol, 2003).

## Qualificação

Em alguns momentos, deseja-se qualificar as declarações sobre as coisas para registrar em que contexto uma declaração é considerada válida (isto é, qual é o seu âmbito de validade, ou abrangência) ou de qual fonte a declaração é.

Em **Topic Maps**, há uma característica para isto: *scope*. Quando uma declaração é feita em um topic map (na forma de um nome, uma ocorrência ou uma associação), um contexto (*scope*) ou abrangência é lhe sempre anexado. Por omissão, todas as declarações de tópicos não são restringidas quanto a contexto, o que significa que não há limites para a validade de tais declarações. Tópicos podem ser adicionados com contexto para restringir em que circunstâncias eles são considerados válidos. Por exemplo:

- Topic Maps são chamados *topic maps* em Inglês, mas em Português chama-se *mapa de tópicos*. Este é o melhor exemplo para a criação de um único tópico com dois nomes, um válido no contexto *Inglês* e outro válido no contexto *Português*;
- Ocorrências do tipo *opinião* exprimem pontos de vista sobre um dado tema. Cada

opinião pode ser associada a um tópico representando a pessoa que forneceu tal opinião, limitando-se o seu âmbito de validade desse tópico;

- Há um grupo de pessoas que acreditam que Ayrton Senna foi o melhor piloto de Formula 1 de todos os tempos. Isto pode ser representado adicionando-se associações *melhor piloto* entre Ayrton Senna e Formula 1, contextualizando isto com um tópico representando este grupo de pessoas.

Em RDF não há uma característica para definir isto, excepto que literais podem ter um identificador de linguagem<sup>9</sup> anexado a eles, o que de alguma forma é um tipo de qualificação. Isto é possível de se obter através de reificação, contudo, reificação em RDF é algo inadequado em casos práticos (como dito anteriormente).

O problema chave aqui é que as declarações em RDF não possuem identidade, o que significa que é impossível fazer recursos que os representam (sem mudança nas declarações). Visto que o modelo não suporta directamente a qualificação, o suporte a qualificação não pode ser adicionado através de reificação. Este é uma das diferenças fundamentais entre Topic Maps e RDF, a qual tem frustrado todas as tentativas de modelar Topic Maps em RDF, de uma maneira natural.

### Possibilidade de fusão

Finalizando esta comparação entre RDF e Topic Maps, pode-se ainda acrescentar dois importantes pontos: contexto (*scope*) e união (*merge*) (Pepper, 2002). Com o conceito de contexto, Topic Maps resolve o problema de validade contextual. Tanto nomes, como ocorrências e papéis de actuação em associações podem ser validados através de seu contexto. Além disso, o contexto permite um suporte multi-língua à Topic Maps.

No que diz respeito à união (*merge*), Topic Maps foi projectado para tornar fácil o processo de fusão de mapas de tópicos, acrescentando mais um diferencial em relação a RDF.

#### 3.5.5 Por que usar Topic Maps como norma?

As subsecções anteriores apresentaram uma série de diferentes características entre Topic Maps e RDF. Pela comparação apresentada, percebe-se que RDF e Topic Maps são similares, ainda que diferentes. Em algumas áreas, eles correspondem-se, enquanto em outras eles são totalmente diferentes. As diferenças fundamentais parecem ser:

- as formas em que as URIs são utilizadas para identificar as coisas (uma forma em RDF, duas formas em Topic Maps) (secção 3.5.4);
- a existência de três tipos de declaração em Topic Maps e somente um em RDF (secção 3.5.4);

---

<sup>9</sup>Um identificador de linguagem (*language identifier*) é uma string conforme RFC 3066 (Harald Alvestrand, 2001), tal como “en-uk” ou “pt-br”.



- as abordagens tomadas para reificação (subsecção 3.5.4) e qualificação (subsecção 3.5.4) de declarações.

Estes três problemas tornam tecnicamente muito difícil unir **Topic Maps** e **RDF** em uma tecnologia única sem efectuar mudanças profundas (seja em uma, ou em ambas), algo que não seria aceitável por ambas comunidades de utilizadores. O problema com os dois usos para URIs e os três tipos de declarações podem ser facilmente resolvidos; porém o problema da reificação e da qualificação das declarações não pode ser resolvido sem uma mudança na representação de declarações em **RDF**.

A partir da comparação apresentada, ficou claro que **Topic Maps** possui poder descritivo superior ao de **RDF**, no sentido de que um mapa de tópicos contém mais informação sobre si do que um modelo **RDF**. Sem o conhecimento de vocabulários **RDF** específicos, uma aplicação conhece menos sobre o modelo **RDF** do que sobre o topic map.

A conclusão a que se chega é que **RDF** foi desenhado para fornecer metadados sobre recursos de informação, enquanto que **Topic Maps** foi projectado para fornecer uma visão alto-nível do domínio coberto pelos recursos. Como é justamente este o objectivo do presente trabalho, a adopção de **Topic Maps** é totalmente indicada.

## 3.6 Sumário sobre Topic Maps

**Topic Maps** é um formalismo para representar o conhecimento contido em um conjunto de recursos de informação, organizando-o em *tópicos*. Esses tópicos têm ocorrências em recursos de informação e associações que representam e definem os relacionamentos entre os tópicos. O conceito, ou tema, representado por cada tópico pode ser inferido ao examinar as associações entre tópicos e os documentos correspondentes identificados pelas ocorrências ligadas ao tópico. Desta forma, um topic map fornece-nos um ponto de vista sobre uma colecção de recursos (Pepper, 2000), disponibilizando conhecimento sobre determinado assunto, que aqui foi designado por domínio de aplicação.

Historicamente, o formalismo **Topic Maps** (uniformizado universalmente através da norma ISO 13250 (Biezunsky, Bryan, and Newcomb, 1999)) foi definido para facilitar a fusão de diferentes esquemas de índices. Um formato comum para a anotação, usado para a indexação, é um passo crucial em direcção ao objectivo da interoperabilidade entre esquemas de índices. O que é necessário ainda é a interoperabilidade semântica. Enquanto que a especificação **Topic Maps** garante interoperabilidade sintáctica, ontologias garantem interoperabilidade semântica.

Se forem construídos a partir de uma ontologia válida, os **Topic Maps** oferecem, às aplicações que os usam, interoperabilidade semântica entre os vários recursos descrito por cada topic map. O próximo capítulo tratará justamente deste ponto: **Topic Maps** dirigidos por Ontologia.



## Capítulo 4

# Ontologias e Topic Maps

*Como a aurora precursora  
Do farol da divindade,  
Foi o vinte de setembro  
O precursor da liberdade.*

-

*Mas não basta pra ser livre  
Ser forte, aguerrido e bravo;  
Povo que não tem virtude,  
Acaba por ser escravo.*

-

*Mostremos valor, constância  
Nesta ímpia e injusta guerra,  
Sirvam nossas façanhas  
De modelo a toda terra.*

Hino Riograndense. Letra: Francisco Pinto da Fontoura

Este capítulo tem por objectivo apresentar uma relação entre **Topic Maps** e ontologias. Apesar do seu tamanho (em número de páginas) ser visivelmente reduzido, a importância deste capítulo no âmbito desta dissertação é considerável, pois após a apresentação de ontologias – realizado na Subsecção 2.2.6 – e de **Topic Maps** – visto em detalhe no Capítulo 3 – se faz necessário uma indicação de como eles se associam.

Além de funcionar como um índice para recursos de informação, **Topic Maps** permite expressar conhecimento. Isso trás uma grande vantagem para o projectista de **Topic Maps**, pois uma representação de conhecimento semanticamente rica e correcta dá mais qualidade a um topic map. Devido ao facto de que uma ontologia é uma representação de conhecimento, este capítulo descreve como ela desempenha um importante papel no projecto de **Topic Maps**.

A norma ISO 13250 **Topic Maps** fornece uma especificação que permite representar conhecimento – em particular o conhecimento conceptual – com o qual se distingue recursos

de informação semanticamente. O processo de criação de ontologias aplicado à construção de **Topic Maps** foca precisamente este aspecto: ele enfatiza o projecto conceptual e a construção de topic maps que reflectem correctamente a semântica do conhecimento implícito. O projecto e construção de **Topic Maps** que fornece significativa expressão de conhecimento deve ser baseado em princípios estabelecidos de projectos de engenharia.

A estrutura desse capítulo é dividida em 5 partes: a Secção 4.1 tem como função refrescar os conceitos de ontologias vistos no Capítulo 2. A Secção 4.2 apresenta os passos para a construção de uma ontologia. A Secção 4.3 clarifica o que se entende por ontologia em **Topic Maps**. A abordagem **Topic Maps** dirigidos por ontologias é citado na Secção 4.4. O capítulo encerra com um sumário, na Secção 4.5.

## 4.1 Ontologia relacionada com modelos de dados

Conforme apresentado na Subsecção 2.2.6, uma ontologia é uma especificação ou *formalização de uma conceptualização* (Gruber, 1993) de determinado universo de discurso, ou domínio de conhecimento. Uma conceptualização é um conjunto de conceitos e de relações entre si. Alternativamente, uma ontologia é uma teoria lógica a qual dá uma explicação explícita de uma conceptualização, projectada para ser compartilhada por agentes (humanos ou computadorizados) com objectivos diversos (Guarino and Giaretta, 1995).

Uma ontologia difere de outros modelos de dados porque sua preocupação principal é com os conceitos e os relacionamentos entre si, onde a semântica desses relacionamentos são aplicados uniformemente.

Em uma estrutura de dados típica, os relacionamentos entre dados são específicos e toda interpretação é necessariamente executada por um programa que acede os dados. No caso de uma base de dados, os relacionamentos entre os dados são parcialmente representados por um esquema de dados; todavia, quase toda interpretação é executada por um gestor de base de dados que acede os dados. Um humano, ou outro programa, desprovido de conhecimento da semântica específica da estrutura de dados ou base de dados em particular não tem a mínima ideia do que aquilo representa.

Em uma ontologia, os relacionamentos são definidos formalmente e a semântica de um dado relacionamento é consideravelmente detalhada. Se esses relacionamentos possuem certos nomes apropriados que identificam seu significado, um humano visualizando uma ontologia pode entendê-la directamente; assim como um programa pode assumir a semântica de um dado relacionamento e actuar sistematicamente através de toda a ontologia.

Os principais elementos de uma ontologia são:

- Conceitos (coisas);
- Classe de conceitos;
- Subclasse de conceitos;
- As propriedades destes conceitos;

- Os relacionamentos entre esses conceitos, que podem ser relações binárias ou de aridade superior a 2; também podem ser de qualquer tipo;
- Restrições e regras sobre esses conceitos e as suas relações.

Resumidamente, ontologias actuam como modelos semânticos conceptuais representando um conhecimento comum em um modelo bem-definido, consistente, completo, extensível, reutilizável e modular.

## 4.2 Como construir uma Ontologia

O processo para a construção de uma ontologia é similar ao usado para construir um modelo orientado a objectos (Booch, Rumbaugh, and Jacobson, 1999) ou um diagrama entidade-relacionamento (Teorey, 1999). Eis uma sequência de passos, ilustrados por um exemplo:

1. Primeiramente, define-se um universo de discurso; isso significa simplesmente escolher qual é o domínio que se pretende modelar. Inicie-se com esse processo de definição do domínio pela listagem dos conceitos que se desejam incluir, isto é, os objectos significativos deste mundo em particular. Muitas vezes, esta tarefa implica analisar em profundidade o domínio, o que pode significar ler os documentos sobre este universo em particular de modo a identificar os temas, verbos, objectos e adjectivos encontrados em suas definições.

Por exemplo, para criar uma ontologia sobre o conceito *Dinamização Científica*, devem incluir-se temas tais como *instituição*, *evento*, *cidade*, *escola*, *universidade* e *capital* (no conjunto dos conceitos), além de relações como *organiza* e *é organizado por*, e *XATA*, *Braga* e *Universidade do Minho* (no conjunto de conceitos concretos, físicos/existentes);

2. Após a fase anterior, lista-se como os temas podem se relacionar entre si. Percebe-se que tais coisas como os verbos identificados acima definem as relações entre os conceitos. Enquanto se faz este processo, pode-se descobrir temas que foram esquecidos durante o primeiro passo. Caso afirmativo, adicionam-se estes novos conceitos e relações.

Neste exemplo, o termo *organizar* é um relacionamento entre *evento* e *instituição*. Deseja-se expressar a ideia que *uma instituição organiza um evento*, e que *um evento é organizado por uma instituição*.

3. Com base na análise realizada nos passos 1 e 2, por fim constrói-se a ontologia de forma manual, ou usando uma ferramenta para gestão de ontologias, como por exemplo o Protégé (Protégé, 2005). Nesta dissertação, apresentam-se duas linguagens para a especificação de extracção de topic maps baseados em ontologias (XSTM no Capítulo 7 e XS4TM no Capítulo 8), possibilitando sua extracção automática e evitando a sua edição manual.

Um guia sobre como projectar uma ontologia está disponível em (Gruber, 1995).

### 4.3 Ontologia em Topic Maps

Ao analisar Topic Maps, percebe-se que tópicos e associações podem ser divididos em duas camadas: os que representam conceitos abstractos e os que representam conceitos concretos. A ontologia de um topic map basicamente é definida pelos conceitos abstractos, ou seja, os que serão definidos como os tipos de tópicos, tipos de associações, tipos de papéis de actuação, tipos de nomes, contextos e tipos de ocorrências. Além disso, consideram-se também como parte da ontologia de um topic map: específicos papéis de actuação relacionados a tipos de associação, tipos de ocorrência de determinados tipos de tópicos, regras implícitas de cardinalidade, etc.

A Figura 4.1 dá uma representação esquematizada desta visão. Essa figura apresenta um topic map para o conceito *Dinamização Científica*, que envolve conceitos como *instituição*, *evento* e *cidade*, além de sub-classes destes conceitos, tais como *universidade*, *escola*, *conferência*, *summer school* e *capital*.

Adicionalmente, encontram-se relações entre os conceitos: tal como *organizar*, que relaciona os conceitos *evento* e *instituição*. Nessa associação, os papéis de actuação “*organiza*” e “*é organizado por*” indicam como cada membro da associação relaciona-se com os demais.

Todos estes conceitos – classes, sub-classes (tipos de tópico), relações (tipos de associação) e papéis de actuação – formam a ontologia deste topic map.

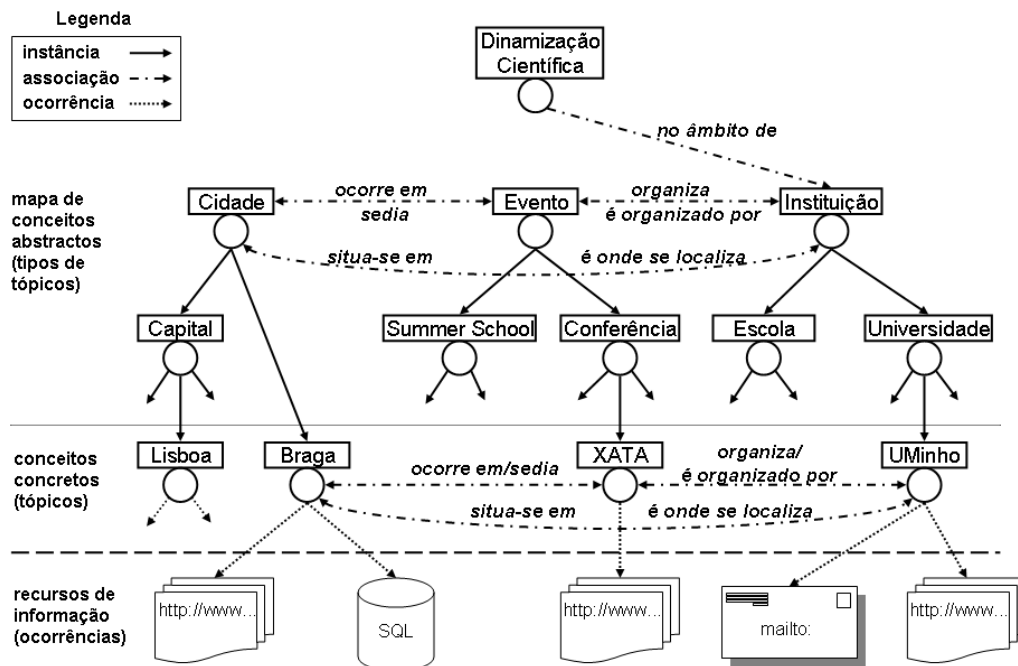


Figura 4.1: Um topic map para o conceito *Dinamização Científica*

Os tópicos restantes formam a base de conhecimento associada à ontologia, os quais compõem um conjunto de objectos de informação que permite organizar os conceitos e indicar os reais recursos de informação associados a eles (um objecto pode ter múltiplas ocorrências nos recursos de informação).

De acordo com o topic map representado na Figura 4.1, os conceitos concretos deste domínio são: *Lisboa*, *Braga*, *XATA* e *UMinho*. É perceptível que nenhum destes conceitos possuem instâncias, ou seja, todos são nodos folhas na árvore hierárquica do topic map.

Para exemplificar a ontologia mencionada no item 2 da Secção 4.2, será usada a sintaxe XTM. Para clarificar o exemplo, apresentam-se os tópicos *Instituição*, *Universidade*, *Universidade do Minho* e *XATA*, assim como a associação entre os dois últimos chamada *organizar*:

```

1 <topicMap>
2   <topic id="instituicao">
3     <baseName>
4       <baseNameString>Instituição</baseNameString>
5     </baseName>
6   </topic>
7   <topic id="universidade">
8     <instanceOf>
9       <topicRef xlink:href="#instituicao"/>
10    </instanceOf>
11    <baseName>
12      <baseNameString>Universidade</baseNameString>
13    </baseName>
14  </topic>
15  <topic id="uminho">
16    <instanceOf>
17      <topicRef xlink:href="#universidade"/>
18    </instanceOf>
19    <baseName>
20      <baseNameString>Universidade do Minho</baseNameString>
21    </baseName>
22  </topic>
23  <topic id="xata">
24    <instanceOf>
25      <topicRef xlink:href="#conferencia"/>
26    </instanceOf>
27    <baseName>
28      <baseNameString>XATA</baseNameString>
29    </baseName>
30  </topic>
31  <topic id="organiza"/>
32  <topic id="organizado-por"/>
33  <association>
34    <instanceOf>
35      <topicRef xlink:href="#organizar"/>
36    </instanceOf>
37    <member>
38      <roleSpec>
39        <topicRef xlink:href="#organiza"/>
40      </roleSpec>
41      <topicRef xlink:href="#uminho"/>
42    </member>
43    <member>
44      <roleSpec>
45        <topicRef xlink:href="#organizado-por"/>
46      </roleSpec>

```

```

47 |         <topicRef xlink:href="#xata"/>
48 |     </member>
49 | </association>
50 </topicMap>

```

Os 3 primeiros tópicos deste topic map na sintaxe XTM apresentam uma relação classe/sub-classe de 3 níveis: *Instituição* (linha 2 a 6) é super-tipo de *Universidade* (linha 7 a 14), que por sua vez é super-tipo de *Universidade do Minho* (linha 15 a 22). O tópico *XATA* (linha 23 a 30) é sub-classe de *Conferência* – o qual não se encontra nesse topic map para simplificá-lo. Os tópicos *organiza* (linha 31) e *organizado-por* (linha 32) são definidos como papéis de actuação em associações, pois os mesmos são referenciados na associação *organizar* (linha 33 a 49), a qual envolve os tópicos *UMinho* e *XATA*. Traduzindo para o bom português, esta associação diz que:

- UMinho organiza a XATA
- XATA é organizado pela UMinho

## 4.4 Topic Maps dirigidos por Ontologia

Esta secção introduz a noção de *Topic Maps dirigidos por Ontologia* (a partir de agora, referenciado apenas como *TMdO*) (Park and Hunting, 2003). Esta abordagem faz parte da tendência em posicionar ontologias no coração dos sistemas informação (Guarino, 1998). Esta tendência determina que o conhecimento é a entidade de principal importância e que as ontologias desempenham o papel central no projecto e operação de sistemas de informação.

### 4.4.1 Como Ontologias se relacionam com Topic Maps

Um formato de anotação comum para designação de índices – objectivo inicial de **Topic Maps** – é um passo crucial em direcção ao objectivo de se conquistar a interoperabilidade entre esquemas de índices; com isso, resta ainda obter-se a interoperabilidade semântica. Enquanto que a especificação **Topic Maps** garante interoperabilidade sintáctica, ontologias asseguram interoperabilidade semântica. Portanto, se **Topic Maps** forem construídos a partir de uma ontologia consistente, eles podem oferecer interoperabilidade semântica.

Topic Maps podem ser gerados a partir de ontologias, como foi brevemente demonstrado na Subsecção 4.2. Com a abordagem *TMdO*, a ontologia torna-se um excelente ponto de partida para a geração de topic maps.

### 4.4.2 Vantagens da abordagem *TMdO*

Um topic map dirigido por uma ontologia torna mais simples o seu processo de criação e manutenção. O projecto de uma ontologia – da qual pretende-se gerar um topic map – deve



ser realizado em separado do processo de construção de um topic map, pois o topic map expresso em XTM pode crescer gradualmente, visando atender os todos temas relevantes para a representação do conhecimento de um domínio em particular. Desta forma, se a ontologia de um dado topic map se mantém inalterada, então somente o mapeamento do domínio para o topic map deve ser realizado.

Uma mudança no mapeamento pode ser ocasionada por uma modificação nos requisitos do processo. Ao separar a ontologia e o topic map, permite-se que as mudanças conceituais possam ser efectivadas independente das demais mudanças. Neste caso, a abordagem *TMdO* oferece as vantagens de uma típica abordagem *loose coupling*<sup>1</sup> (March and Olsen, 1976).

A abordagem *TMdO* disponibiliza o uso de uma grande quantidade de ontologias existentes. As ontologias são o resultado de um investimento significativo, como são os **Topic Maps**. Tal abordagem poupa esforços na construção de um domínio para o qual um trabalho de representação de conhecimento tenha sido realizado previamente.

Outro benefício do uso de ontologias existentes é que muitas destas ontologias têm sido testadas e usadas com sucesso por várias aplicações. Exemplos na área médica incluem as ontologias encontradas nos projectos *Unified Medical Language System* (UMLS)<sup>2</sup> e *Generalised Architecture for Languages, Encyclopaedias and Nomenclatures in medicine* (GALEN)<sup>3</sup>. Com isto, faz sentido o reutilização de porções relevantes destas ontologias para construir topic maps sobre medicina, por exemplo.

A abordagem *TMdO* também tem como vantagem o facto de que as ontologias construídas para a criação de **Topic Maps** podem ser utilizadas para outros fins.

As ontologias são construídas usando linguagens específicas para a representação do conhecimento, enquanto que **Topic Maps** foram criados especificamente para organizar recursos na Internet (isto é, criando uma camada semântica sobre os recursos que permanecem inalterados). Uma ontologia simples, então, pode ser usada em várias aplicações além da geração de **Topic Maps**, como por exemplo: motores de inferência, aplicações de linguagem natural, etc. Desta forma, é recomendado manter a conceptualização do domínio em uma ontologia, deixando a implementação real do domínio para um topic map.

Até o presente momento, não é conhecida nenhuma ferramenta para a geração de topic maps a partir de uma ontologia. Por isso, o extractor automático de topic maps (escritos em XTM) a partir de um conjunto de recursos de informação heterogêneos com base na descrição de uma ontologia subjacente, que se apresenta à frente no Capítulo 8 – designado por *Oveia* – é considerado um contributo inovador desta tese, pois o *Oveia* permite a geração de topic maps dirigidos por ontologias.

---

<sup>1</sup>O conceito de *Loose Coupling* é um conceito fundamental da programação. O código de qualidade deve manter os seus diversos componentes *loosely coupled*, ou seja independentes entre si mas com uma grande coesão interna. Por componentes entende-se classes, assemblies, tudo o que possa ser considerada uma unidades lógica de funcionalidade.

<sup>2</sup>UMLS: <http://www.nlm.nih.gov/research/umls/>

<sup>3</sup>GALEN: <http://www.cs.man.ac.uk/mig/projects/old/galen/brochure.html>

#### 4.4.3 As tendências para a abordagem *TMdO*

Obviamente, é possível criar topic maps de uma forma directa, especialmente se o topic map é pequeno e expressa uma quantidade limitada de conhecimento. Além disso, alguns topic maps não necessitam oferecer muito mais do que uma simples classificação de recursos de informação.

Contudo, se esta-se a construir um índice simples para um documento ou um mapa de navegação sofisticado para um web site, implicitamente está-se a expressar conhecimento. Assim, criando uma camada de conhecimento rica semanticamente (a ontologia), se estará adicionando qualidade significativa para o topic map.

No futuro, a tendência aponta para a geração de topic maps a partir de ontologias, onde o conhecimento será mantido e gerido pela edição das ontologias, as quais poderão ser publicadas em conjunto com seus topic maps. Para aqueles topic maps que não forem gerados a partir de ontologias, ao menos deverão ser projectados com princípios baseados em ontologia para garantir consistência semântica dos seus termos.

Poderão ser encontradas bibliotecas de topic maps, assim como bibliotecas de ontologias para sua geração. Para uma dada ontologia, poderão haver múltiplos topic maps, onde cada um seria a imagem de um mapeamento distinto. Espera-se que ontologias normalizadas se tornem disponíveis para vários domínios de conhecimento.

### 4.5 Sumário sobre Ontologia e Topic Maps

Este capítulo descreveu o relacionamento entre ontologias e Topic Maps. Assim como Topic Maps, ontologias também podem ser usadas para adicionar informação conceptual e semântica em documentos Web e ferramentas. Isso conduz a uma interpretação humana bem próxima do que os projectistas desejam.

Depois de relembrar a definição de ontologia (apresentada na Secção 2.2.6) e após uma breve apresentação sobre como construir ontologias, tendo ainda em consideração a definição de Topic Maps (Secção 2.4.3 e Capítulo 3), pode-se fazer um mapeamento entre uma Ontologia e Topic Maps. Assim, definiu-se o mapeamento abaixo representado:

- Conceito -> Tópico
- Classe -> Tipo de tópico
- Subclasse -> Instância de tópico
- Propriedades das Classes -> Características dos tópicos
- Relacionamento -> Associação
- Axiomas que representam as condições -> ???

A partir deste mapeamento, percebe-se que quase todos os principais elementos de uma ontologia podem ser mapeados para elementos de **Topic Maps**. Entretanto, a norma **Topic Maps** não possui mecanismos para definir os axiomas que representam as condições a serem impostas sobre uma ontologia, pois uma ontologia necessita de um mecanismo que possibilite a validação de sua representação, através de restrições semânticas. Essas restrições podem garantir que uma ontologia realmente representa o que o projectista tinha em mente.

Quando a ontologia está representada num topic map, deve-se ter mecanismos de validação do conhecimento nele representado. Para isso, o próximo capítulo introduz as motivações para essa validação semântica sobre topic maps e o Capítulo 9 apresenta a linguagem **XTche**, que foi desenvolvida para suprir essa carência encontrada no domínio dos **Topic Maps**.



## Capítulo 5

# Validação Semântica em Topic Maps

*Quero gaita de oito baixos pra ver o ronco que sai  
Botas feitio do Alegrete e esporas do Ibirocaí  
Lenço vermelho e guaiaca  
compradas lá no Uruguai  
Pra que digam quando eu passe  
saiu igualzito ao pai*

César Passarinho

Um topic map, assim como uma base de dados ou um documento XML, fornece um modo de estruturar conhecimento; para se acreditar na qualidade (validade) das informações descritas, são necessários métodos e notações para especificar regras semânticas a que o topic map deve obedecer. Porém, os **Topic Maps** ainda não possuem uma forma eficiente de descrever as regras que os tópicos e associações devem cumprir para serem considerados válidos semanticamente.

Este capítulo discute a necessidade de um mecanismo para a descrição de restrições em **Topic Maps**, que permita confirmar a validade de um topic map de acordo com as intenções do projectista. Consequentemente, o capítulo mostrará a necessidade de se ter um sistema que permita a especificação de restrições sobre topic maps. Serão apresentados exemplos de restrições que fornecerão algumas ideias do grau de abrangência deste sistema.

### 5.1 Restrições

Dada uma especificação (de uma estrutura de dados ou de uma operação), uma restrição (do inglês, *constraint*) é uma expressão lógica que condiciona os possíveis valores que uma variável pode tomar.

As restrições também podem ser vistas como um acordo entre fornecedores de um serviço e os receptores do produto gerado por este serviço. O acordo define as condições em que os serviços serão prestados e a especificação do resultado do serviço que é fornecido.

Por exemplo, a frase “*o livro está sobre a mesa*” relaciona dois objectos: o livro e a mesa. A adição de outro objecto relacionado com a mesa, digamos uma revista, pode ser descrita pela declaração: “*a revista está sobre a mesa*”. Se é importante declarar que a posição entre o livro e a revista não é arbitrária, pode-se usar uma restrição: “*a revista deve estar à esquerda do livro*”. Neste caso, a restrição impõe que a variável *posição* não pode tomar valores como *à direita*, *sobre*, etc. A partir de agora, dada uma configuração de uma mesa, é possível dizer *se ela é válida ou não*; isto é, se a configuração da mesa satisfaz a restrição ou não.

As restrições podem ser aplicadas a especificações de um qualquer domínio.

O conjunto de frases válidas de uma linguagem formal pode ser restringido através de condições contextuais colocadas nas gramáticas de atributos. O processo de prova em programação lógica também pode ser controlado adicionando-se restrições aos predicados.

Quando se trata de documentos anotados, algumas linguagens de domínio específico, tais como Schematron (Dodds, 2001) e XCSL (Jacinto et al., 2002a), podem completar as definições de tipos (DTD ou XML Schema (Duckett et al., 2001)) de documentos XML com restrições semânticas. Estas linguagens de domínio específico permitem descrever o condicionamento ou limitações impostos por cada problema de uma maneira directa, clara e simples. Além disto, elas permitem a derivação de um programa para automatizar a tarefa de validação. O validador semântico derivado verificará cada documento XML, mantendo silêncio quando as restrições são satisfeitas e reportando erros quando as condições contextuais são violadas.

Uma definição de restrição no âmbito de Topic Maps encontrada na literatura tem-se abaixo (Grønmo, 2000):

Uma maneira artificial de limitar (restringir) ou controlar um ou mais valores de propriedades dos nodos na área coberta por um topic map.

## 5.2 A necessidade de validação em Topic Maps

A publicação da norma Topic Maps (Biezunsky, Bryan, and Newcomb, 1999) gerou um grande interesse na comunidade científica. Para muitos, isto não foi uma surpresa, visto que esta norma permite a criação de estruturas navegacionais baseadas na representação de relacionamentos entre temas e ligação de informações adicionais a estes temas.

Conforme apresentado no Capítulo 3, um topic map é um conjunto organizado de tópicos (representação formal de temas), com links para recursos de informação indexados por estes tópicos (ocorrências) e relacionamentos semânticos entre tópicos (associações). Consequentemente, um topic map pode ser visto como uma infra-estrutura que permite a descrição de informação, organização de conhecimento e o acesso aos recursos de informação.

Por exemplo, um topic map permite organizar o conhecimento entre duas pessoas – o aluno Giovani Librelotto e professor Pedro Henriques – dizendo que entre eles há uma associação ou ligação – no caso, uma *orientação* – onde o primeiro desempenha o papel de *orientando* e o segundo, o papel de *orientador*.

Conforme apresentado nos dois capítulos anteriores, os **Topic Maps** fornecem diferentes mecanismos, permitindo uma rica expressividade do conhecimento, através:

- da gestão de vários nomes para cada tópico;
- da gestão dos apontadores (ocorrências) entre os tópicos e documentos externos que estão indexados;
- da gestão de relações semânticas (associações), hierárquicas ou não, entre tópicos;
- da capacidade de classificar tópicos com outros tópicos, suportando uma multi-classificação;
- de um mecanismo de filtragem (contexto) de objectos, permitindo um ajuste dos tópicos de acordo com as necessidades específicas dos utilizadores;
- de um mecanismo de identificação universal dos temas descritos pelos tópicos (PSI ou Indicadores de Temas Públicos) permitindo uma identificação única de um tema em diferentes **Topic Maps**; um suporte crítico para fusão;
- da capacidade de intercambiar e fundir **Topic Maps**.

Devido a esta riqueza de mecanismos, um topic map possui a capacidade de expressar e organizar o conhecimento de um domínio qualquer. Contudo, já não permite facilmente limitar os valores concretos que tópicos, associações e ocorrências (respeitando a forma, ou estrutura) possam tomar situações de aplicações reais. Além disso, a verificação da semântica deste mesmo topic map torna-se complicada, na medida em que este domínio poderá conter um numeroso conjunto de temas, relações e apontadores para recursos externos.

Além disso, os topic maps podem ser projectados e editados por diversos pesquisadores e pessoas em geral, o que pode acarretar uma certa perda do controle da sua consistência, a qual é, como se define acima, difícil de ser verificada. Consequentemente, os topic maps podem facilmente tornar-se bastante complexos. Ou seja, ao criar e manter topic maps sem ajuda de ferramentas adequadas, podem introduzir-se contradições; inconsistências essas que devem ser detectadas e eliminadas.

A norma **Topic Maps** tem um conjunto limitado de mecanismos para a definição de restrições no mapa de tópicos. Quase nenhum destes, entretanto, pode ser usado para restringir a semântica definida pelo utilizador.

Remontando o exemplo anterior, em que se introduziu a associação *orientação*, é claro que seria importante afirmar que para serem consideradas válidas, as associações do tipo

*orientação* têm um papel de actuação *orienta*, que devem necessariamente ser desempenhado por um único tópico do tipo *professor*, assim como um papel de actuação *orientado* que deverá necessariamente ser desempenhado por um ou mais tópicos do tipo *aluno*. Os tipos de tópicos *aluno* e *professor* devem ser sub-tipos do tópico *pessoa* e ambos possuem um nome do tipo *nome-completo* e uma ocorrência do tipo *e-mail* que contém o carácter “@” e está inserida no contexto *internet*.

Para evitar as inconsistências comuns na edição de topic maps, os utilizadores necessitam de um mecanismo que permita ajudá-los a manter sua solidez. Assim, uma ferramenta para esta finalidade deve ser capaz de guiar o utilizador na construção de topic maps válidos e informá-lo dos eventuais erros encontrados nos topic maps que vão sendo criados.

No contexto dos **Topic Maps**, isto significa que o projectista da ontologia e o editor do topic map concordam com as regras que definem a constituição de um topic map válido e, por conseguinte, consistente.

### 5.3 O que já se pode restringir?

A intenção esta secção é apresentar um conjunto de restrições que, até o princípio da pesquisa sobre esta tese, eram possíveis de ser aplicados a topic maps. A secção divide-se em duas: a primeira parte, trata o que é abordado pela norma **Topic Maps**; a segunda parte, aborda as restrições que o uso de XML trouxe ao ser adoptado para a representação de topic maps.

#### 5.3.1 Com a norma Topic Maps

A norma **Topic Maps** não restringe o que pode ser representado em topic maps, assim como não restringe o carácter do processamento que pode ser realizado por uma aplicação. A cláusula *Note 50* da publicação ISO/IEC 13250 **Topic Maps** (Biezunsky, Bryan, and Newcomb, 1999) garante que os topic maps podem ser entendidos por qualquer aplicação, e que a informação expressa num topic map, de acordo com a sua sintaxe, será preservada por aplicações de leitura e escrita, a menos que os utilizadores tenham a necessidade de alterar este mapa de tópicos.

A norma define, em seu contexto, as restrições sintácticas. No entanto, poucas restrições semânticas estão definidas; e elas estão num nível diferente das que serão apresentadas nesta dissertação.

O tipo de restrição encontrado na norma **Topic Maps** refere-se à restrições aos nomes dos tópicos (*The topic naming constraint*) (Mason and Desautels, 2002). Isso quer dizer que dois temas distintos não podem ter o mesmo nome quando estão no mesmo contexto. A norma define que, quando isso acontecer, a aplicação que está a processar o topic map deve fundir estes tópicos num único.

Por exemplo, dois tópicos não podem ter o nome *Ténis* no mesmo contexto. Neste caso, os dois tópicos devem ser fundidos pela aplicação, sem levar em conta que eles se referem



a um tipo de calçado e a um desporto, os quais são seguramente dois temas diferentes.

Como pode-se perceber, a norma **Topic Maps** (Biezunsky, Bryan, and Newcomb, 1999) é muito explícita sobre o facto de que não há limites para o seu uso. Um exemplo disto são os recursos de informação e as suas relações com os objectos de **Topic Maps**, pois a norma não impõe restrições sobre a natureza das informações dos objectos que podem ser especificados como ocorrências de tópicos, nem sobre as notações de endereçamento usadas para referenciar tais ocorrências.

### 5.3.2 Em XML Topic Maps

Como já foi mencionado anteriormente, a norma **Topic Maps** não especifica restrições semânticas. Não há praticamente nada na norma que auxilie o editor na verificação de como os tópicos serão interpretados no topic map.

Os tipos de restrições definidas pelo uso de XML para representar topic maps são (Grønmo, 2000):

**Restrições Estruturais:** No momento em que o formato de intercâmbio para **Topic Maps** foi definido em XML, um DTD foi criado (Pepper and Moore, 2001b). Isto tornou possível a restrição explícita do uso de elementos e atributos de **Topic Maps**. Assim, o DTD de XTM possui a capacidade de validar os topic maps sintacticamente.

**Restrições Implícitas:** Restrições que não estão explicitamente especificadas pela norma, mas estão implícitas pelo facto de que o XML é usado como formato de intercâmbio. Por exemplo, o identificador de um tópico deve estar de acordo com as restrições aplicadas aos identificadores de elementos de XML: não devem existir dois identificadores iguais.

Contrastando com XML, a norma **Topic Maps** não tem mecanismo (até o presente momento) para aplicação de restrições. XML possui algumas linguagens especiais para a definição de restrições sintácticas em documentos, nomeadamente DTD (*Document Type Definitions*), XML-Schema e Relax-NG, entre outras.

Adoptando-se o formato XTM para a representação concreta de **Topic Maps**, a validação sintáctica é garantida por um parser XML porque a estrutura de um documento XTM é definida por um DTD (Pepper and Moore, 2001b). Contudo, sabe-se que a validação estrutural não significa uma completa correcção – a semântica também deve ser garantida.

Usando XML Schema para a validação de XTM, ao invés de DTD, aprimora-se o processo de validação porque alguns requisitos de semântica (domínio, número de ocorrências, etc.) podem ser adicionados à especificação estrutural; parsers XML tratam assim desta tarefa. Contudo, outros requisitos de semântica permanecem não-especificados. Assim, torna-se necessária uma linguagem de especificação que permite a definição de esquema e restrições em uma família de **Topic Maps**.

Para a definição de restrições semânticas em documentos XML usa-se, principalmente, linguagens de domínio específico para este fim, tais como Schematron (Dodds, 2001) e

XCSL (Jacinto et al., 2002a) (Jacinto et al., 2002b). São linguagens simples e pequenas, desenvolvidas para a especificação de condições contextuais que avaliam o valor textual de elementos e atributos em documentos XML, de uma mesma família (pertencentes ao mesmo esquema).

A Figura 5.1 apresenta a visualização textual de um topic map na sintaxe XTM e a sua visão gráfica, na forma de uma rede semântica. Como pode ser observado, a sintaxe XTM organiza os tópicos sequencialmente, ou seja, uns abaixo dos outros (segundo a ordem linear de um texto). Em XTM, a ordem dos tópicos e associações é irrelevante, pois as relações entre os elementos `<topic>` e `<association>` são definidas através de referências `xlink:href` para os seus identificadores.

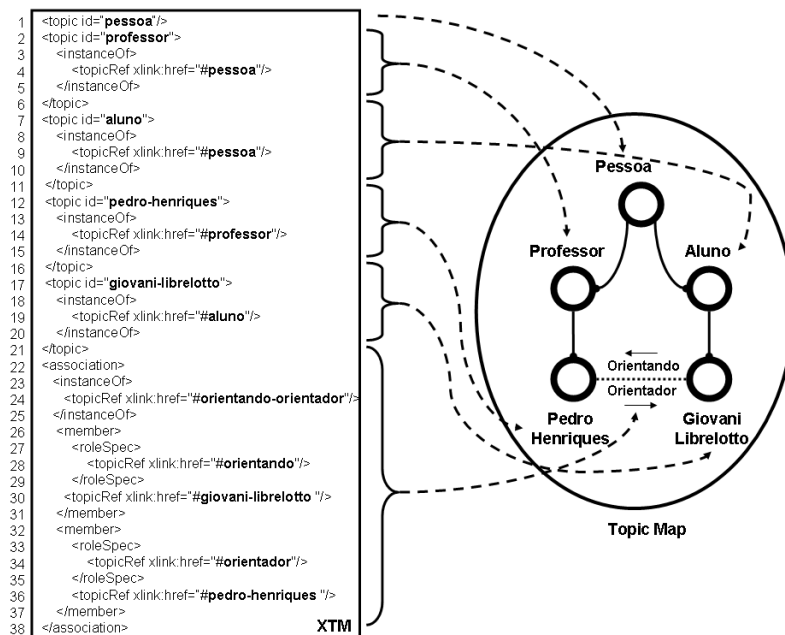


Figura 5.1: Representação de um topic map na sintaxe XTM e sua visão em forma de grafo.

Por exemplo, a referência `xlink:href` encontrada na linha 9, realizada pelo tópico *aluno* (linhas 7 a 11) ao seu tipo de tópico *pessoa* (linha 1), relaciona-os de forma hierárquica (classe/instância) independente de suas posições físicas no documento XTM.

Contudo, as referências `xlink:href` não são validadas semanticamente pelo esquema de XTM, definido por seu DTD. Há então a necessidade de validar a rede semântica (o topic map da Figura 5.1) e não a sintaxe XTM. Ou seja, verificar se os membros da associação do tipo *orientando-orientador* são realmente instâncias dos tópicos *professor* e *aluno*. Mesmo com a utilização de linguagens como XCSL e Schematron, tal validação não pode ser obtida.

A solução então é uma linguagem que preocupa-se com as relações entre os nodos da rede semântica, e não com a estruturação do documento XML. Para esse fim, a próxima secção introduz a proposta da ISO para uma linguagem de especificação de restrição em

**Topic Maps**, a qual servirá de base para a concretização da linguagem **XTche** (definida no Capítulo 9).

## 5.4 ISO 19756: Topic Map Constraint Language – TMCL

*Topic Map Constraint Language* (TMCL) (Moore, Nishikawa, and Bogachev, 2004) não é, na verdade, uma linguagem concreta, mas sim é uma lista de requisitos para a criação de uma linguagem formal para a definição de esquemas e restrições sobre **Topic Maps**. TMCL especifica restrições em topic maps em conformidade com o modelo de dados de **Topic Maps** (*Topic Maps Data Model* – TMDM) (Garshol et al., 2003). A futura linguagem TMCL irá fornecer a base para uma linguagem formal para restrições, incluindo uma sintaxe.

TMCL também poderá ser utilizada na otimização do armazenamento de topic maps e nas consultas TMQL (Garshol and Barta, 2005). Isso pode auxiliar na geração de interfaces intuitivas ao utilizador, para a criação e manutenção de topic maps.

TMCL possui uma série de requisitos (Nishikawa, Moore, and Bogachev, 2004), dos quais destacam-se os seguintes:

**Sintaxe XML:** A linguagem TMCL deve possuir somente uma sintaxe normalizada XML;

**Modelo e sintaxe:** TMCL deve definir: (a) uma sintaxe para expressar restrições; (b) um modelo para representação interna destas restrições; (c) o comportamento da validação TMCL;

**Restrições em características de tópicos:** A linguagem deve ser capaz de expressar restrições em características de tópicos: nomes, ocorrências e identificadores de tema;

**Conjunto de caracteres:** O conjunto de caracteres para TMCL deve ser Unicode.

Em relação a sintaxe, TMCL deverá ter uma sintaxe amigável baseada em XML, que talvez possa ser baseada em XTM.

Os resultados de uma validação devem ser a confirmação de que o topic map satisfaz o conjunto de restrições fornecido. Caso alguma restrição não é satisfeita, então mensagens de erro devem ser mostradas. Essas mensagens de erro podem também incluir advertências (*warnings*), dependendo da regra de validação a ser aplicada.

TMCL não prevê a especificação de restrições sobre os identificadores de tópicos, assim como em que situações cada tipo de restrições pode ser aplicado. Além disso, TMCL não vai definir vocabulários explícitos para certos domínios; contudo, pode ser usada para propósitos ilustrativos em cenários de uso.

O progresso do desenvolvimento de TMCL está parado devido as dependências na família de normas de **Topic Maps** (Garshol, 2002b; Biezunsky, Bryan, and Newcomb, 2003) e pode também ficar dependente de OWL (Bechhofer et al., 2002) e dos tipos de dados de XML-*Schema* (Biron, 2001). Conforme essas normas avançam, o mesmo acontece com TMCL,

pois com as mudanças nos modelos de dependência de TMCL – modelo de dados de **Topic Maps**, especificação da sintaxe XTM e TMQL – a compatibilidade e a integração entre estas deve ser sempre garantida.

No momento, o projecto de *Topic Maps Constraint Language* está sob desenvolvimento pelos membros do comité *ISO/IEC JTC 1/SC 34* (Moore, Nishikawa, and Bogachev, 2004). Como se disse no início, a TMCL não é ainda uma linguagem concreta; é apenas uma lista de requisitos. O passo seguinte será a definição de uma linguagem que implemente todos os seus requisitos, e daí tem sido tomada como um dos objectivos deste trabalho de doutoramento.

O mecanismo de restrições TMCL abrange todos os objectos<sup>1</sup> encontrados num topic map. Porém, alguns destes objectos devem possuir um conjunto maior de restrições. Dentre todos, o mais importante é a associação, pois ela representa a semântica principal de um topic map: as relações entre tópicos.

Com a lista dos tipos de restrições encontrada nas próximas subsecções (Nishikawa, Moore, and Bogachev, 2004), pretende-se apresentar os tipos de restrições que devem ser definidos. As restrições apresentadas são atómicas e podem ser combinadas.

#### 5.4.1 Restrições em Associações

O tipo de associação é o ponto de partida para a descrição de uma restrição para um conjunto de associações. As **associações** de um dado tipo terão, provavelmente, algumas regras rígidas, especificando como essas associações devem ser estruturadas, especialmente como os papéis de actuação e os tópicos participantes devem ser combinados.

Assim, as **associações** podem ser restringidas usando os seguintes critérios:

- O tipo da associação — por exemplo: o tópico *orientação* pode somente ser utilizado como tipo de associação;
- O contexto (*scope*) onde a associação está inserida — por exemplo: toda associação do tipo *orientação* deve estar inserida no contexto *educação*;
- Os papéis de actuação permitidos numa associação — por exemplo: toda associação do tipo *orientação* deve conter os papéis *orientador* e *orientando*;
- A cardinalidade dos papéis numa associação — por exemplo: toda associação do tipo *orientação* deve conter um papel *orientador* e um ou mais papéis *orientando*;
- O tipo dos papéis de actuação permitidos numa associação — por exemplo: toda associação do tipo *orientação* deve conter papéis do tipo *funções-em-universidade*, que é tipo dos tópicos *orientador* e *orientando*;

---

<sup>1</sup>Por objectos em **Topic Maps** entendem-se todos os elementos básicos designados por TAO (Pepper, 2000) – tópicos, associações, ocorrências – e os elementos a eles associados – nomes, contextos, papéis de actuação em associações, actores em associações, etc.

- Os tópicos participantes — por exemplo: toda associação do tipo *estuda-na-UMinho* deve conter o tópico participante *Universidade do Minho*;
- O tipo dos tópicos participantes — por exemplo: toda associação do tipo *orientação* deve conter tópicos dos tipos *professor* e *aluno*;
- A cardinalidade dos tópicos participantes — por exemplo: toda associação do tipo *orientação* deve conter um tópico do tipo *professor* e um ou mais tópicos do tipo *aluno*;
- A combinação de tipos de tópicos e papéis de actuação, permitindo asserções sobre a cardinalidade;
- Dependências entre associações.

Como exemplo de uma restrição envolvendo dependências, pode-se usar as associações do tipo *conter*, as quais devem conter exactamente dois papéis de actuação, onde um dos papéis deve ser *contém*, enquanto que o outro papel deve ser *está contido*. Se o tópico que está a participar com o papel de actuação *contém* é uma instância de *país*, então o tópico que participa com o papel *está contido* deve ser uma instância de *cidade* ou *estado* (*província*). Por outro lado, se o tópico que participa com o papel *contém* é uma instância de *estado*, então o tópico que participa com o papel *está contido* deve ser uma instância de *cidade*.

Outro exemplo poderia ser: as associações do tipo *fronteira* devem conter exactamente dois papéis de actuação. Se a associação está no contexto *países*, então os tópicos participantes devem ser instâncias de *país*. Se a associação está no contexto *continentes*, então os tópicos participantes deverão ser instâncias de *continente*.

### 5.4.2 Restrições em Tópicos

O tipo de tópico é o ponto de partida para descrever uma restrição sobre um conjunto de tópicos. Tópicos de um dado tipo muito provavelmente terão regras rígidas sobre como as características desses tópicos deverão ser especificadas. Como características de tópicos, entende-se os seus nomes, contextos, indicadores de temas e ocorrências.

Os tópicos podem ser restringidos usando os seguintes critérios:

- Os tipos de tópicos — por exemplo: o tópico *aluno* somente pode designar um tipo de tópicos;
- O padrão das características de tópicos — por exemplo: os nomes do tipo *tutorial* dos tópicos do tipo *curso* devem conter a palavra “*Tutorial*”;
- A cardinalidade das características de tópicos — por exemplo: tópicos do tipo *aluno* devem ter um indicador de tema, dois nomes e três ocorrências;

- A combinação válida de nomes de tópico — por exemplo: tópico do tipo *aluno* deve ter um nome do tipo *nome-completo*, um nome do tipo *nome-de-tratamento* e um nome do tipo *letras-iniciais*;
- O padrão dos nomes de tópico — por exemplo: os nomes do tipo *nome-de-tratamento* dos tópicos do tipo *professor* deve sempre iniciar com a sequência de caracteres “Prof. Dr.”;
- O tamanho dos nomes de tópico — por exemplo: o nome do tipo *letras-iniciais* possui apenas 3 caracteres;
- As ocorrências de tópicos — por exemplo: todo tópico do tipo *aluno* deve ter ocorrência do tipo *e-mail*;
- Os indicadores de temas (*subject indicators*) — por exemplo: tópicos do tipo *país* devem ter indicadores de temas que referenciam a URI <http://www.topicmaps.org/xtm/1.0/country.xtm>;
- Os tópicos usados para especificar contexto (*scope*) — por exemplo: tópicos do tipo *educação* devem ser utilizados apenas como contexto;
- As instâncias de um tópico — por exemplo: o tipo de tópico *cores-primárias* deve possuir somente as instâncias *azul*, *amarelo* e *vermelho*;
- As associações onde os tópicos podem participar — por exemplo: tópico *orientador* só pode participar em associações do tipo *orientação*;
- As combinações das características de tópicos — por exemplo: tópicos do tipo *aluno* deve possuir um indicador de tema, um nome do tipo *nome-completo* que está no contexto *português*, uma ocorrência do tipo *e-mail* que contém o caracter “@” e uma ocorrência do tipo *idade* que possui 3 caracteres.

Como exemplo, cita-se os tópicos que são instâncias do tópico *país*, os quais devem ter exactamente uma ocorrência que é uma instância do tipo *bandeira*. Um tópico que é uma instância do tipo *país* também deve participar exactamente numa associação binária do tipo *contém*, onde o outro participante deve ser uma instância de *continente*.

Outro exemplo, podia ser: tópicos com nomes no contexto *português* devem ter no mínimo uma ocorrência do tipo *descrição*, também no contexto *português*.

### 5.4.3 Restrições em Ocorrências

As *ocorrências* podem ser restringidas pelo tipo de recurso de informação a que elas estão associadas e pela maneira como esse recurso é endereçado.

As *ocorrências* podem ser restringidas usando os seguintes critérios:

- Os tipos de tópicos onde cada ocorrência pode aparecer — por exemplo: ocorrências do tipo *numero-aluno* só podem ser encontradas nos tópicos do tipo *aluno*;

- Os tipos de ocorrências — por exemplo: o tópico *e-mail* somente pode ser utilizado como tipo de ocorrência;
- Os contextos (*scope*) onde as ocorrências podem ser inseridas — por exemplo: ocorrências do tipo *e-mail* só podem ser encontradas dentro do contexto *internet*;
- O padrão das ocorrências — por exemplo: toda ocorrência do tipo *website* deve iniciar pela sequência de caracteres “**http://**”;
- O tamanho das ocorrências — por exemplo: toda ocorrência do tipo *idade* deve ter no máximo 3 caracteres;
- O formato do recurso de informação endereçado — por exemplo: GIF, HTML, etc;
- A localização do recurso — por exemplo: interno, localizado em um país específico;
- A notação de endereçamento — por exemplo: XLink, HyTime, etc;
- O tipo de endereço — por exemplo: URL, URI, etc;
- As propriedades do endereço — por exemplo, pelo protocolo: http, ftp, etc.

Por exemplo, ocorrências do tipo *bandeira*, no contexto *imagem JPG*, devem referenciar as bandeiras encontradas em <http://geocities.yahoo.com.br/aimorecoelho/bandeiras.html>.

Como outro exemplo, define-se que as ocorrências do tipo *descrição* devem apontar apenas para *documentos XML* como recursos de informação.

#### 5.4.4 Acréscimos à lista de Requisitos de TMCL

Nesta secção são apresentados alguns acréscimos aos requisitos definidos pela ISO, os quais foram submetidos e aceites pelo comité a partir dos resultados obtidos durante o estudo do estado da arte.

O princípio do trabalho de investigação sobre restrições em **Topic Maps** teve como base os requisitos que a ISO impunha em uma lista (Nishikawa and Moore, 2003a) – datada de 04 de Abril de 2003 – que apresentava uma visão geral dos tipos de restrições que deveriam ser aplicados sobre os elementos de um topic map.

Mais tarde, no dia 20 de Outubro de 2003, foi publicado uma lista de requisitos para TMCL incluindo casos de uso para cada tipo de restrição a ser adoptada (Nishikawa and Moore, 2003b). Essa lista estendia a publicação realizada em Abril do mesmo ano com a inclusão de situações reais onde uma validação em topic maps seria necessária.

Foi então realizada uma análise detalhada das situações apresentadas no documento em questão. Chegou-se a conclusão que esta lista não seria a definitiva, pois havia restrições que não faziam sentido (Nishikawa and Moore, 2003b):

**4.1.2.U3(c) – Topic of type T must have a subject indicator in scope S:** esta restrição não fazia sentido pois indicadores de tema não possuem contexto (*scope*);

**4.1.2.U3(d) – Topic of type T must have a subject indicator of type O:** esta restrição não fazia sentido pois indicadores de tema não possuem tipo;

As conclusões acima foram sugeridas ao comité *ISO 19756 Topic Maps Constraint Language* (através da lista de discussão de TMCL<sup>2</sup>) e ao comité *ISO 13250 Topic Maps* (através da lista de discussão SC34WG3<sup>3</sup>). Lars Marius Garshol, um dos maiores experts em **Topic Maps**, confirmou a validade das sugestões na lista de discussão de TMCL (Garshol, 2004d) em 26 de Fevereiro de 2004. Posteriormente, as sugestões apresentadas foram aceites pelo comité e foram retiradas da actual lista de requisitos de TMCL (Nishikawa, Moore, and Bogachev, 2004) publicada em 16 de Outubro de 2004.

Actualmente, novos comentários estão em discussão entre os utilizadores de tais listas. No que diz respeito a temas propostos a TMCL, originados a partir desta dissertação, a seguinte sugestão foi feita a fim de alterar uma vez mais os requisitos de TMCL (Nishikawa, Moore, and Bogachev, 2004):

**4.1.2.U4.(c) – Topic T can be used for typing subject indicator:** tal como a segunda sugestão anterior, esta aqui não possui função alguma, devido ao facto que indicadores de tema não possuem tipo.

Além disso, uma nova secção foi sugerida a essa lista, referindo-se especificamente ao tratamento de tipos de nomes de tópicos, da mesma forma como ocorre com os tipos de ocorrências (Secção 4.1.3. *Constraint by occurrence type*). Desta forma, as regras propostas foram as seguintes:

**Lugares válidos onde os nomes podem aparecer:** Por exemplo: nome do tipo *Apelido* deve ser somente característica de tópicos do tipo *Pessoa*;

**Tipos de nome e contextos válidos:** Por exemplo: nome do tipo *Apelido* deve estar inserido no contexto *Nome Completo*;

**Padrão de um tipo de nome:** Por exemplo: nome do tipo *Email* deve conter o caracter @.

Esse conjunto de propostas foi submetido ao comité de TMCL, porém até o momento, não foi tomado alguma decisão quanto ao seu aproveitamento na próxima publicação dos requisitos dessa linguagem. Contudo, tais ideias foram adoptadas pela linguagem de especificação de restrições em **Topic Maps** chamada XTche, que será descrita no Capítulo 9.

<sup>2</sup>tmcl-wg – Topic Map Constraint Language – <http://www.isotopicmaps.org/mailman/listinfo/tmcl-wg>

<sup>3</sup>sc34wg3 – Discussion of ISO/IEC 13250 Topic Maps – <http://www.isotopicmaps.org/mailman/listinfo/sc34wg3>



## 5.5 O processo de validação de Topic Maps

Restrições e validação estão relacionadas. Validação é o acto de conferir a autenticidade a um objecto, afirmando que este foi verificado e que está de acordo com um padrão, ou seja, um conjunto de regras que definem sua forma e conteúdo; as restrições são as regras que limitam o conteúdo. O processo de validação é necessário em todos os tipos de processamento de informação.

O processo de validação de Topic Maps normalmente consiste em dois passos, que são:

1. verificação da estrutura, ou forma (reconhecimento sintáctico);
2. verificação de conformidade com as regras semânticas, isto é, o respeito pelas restrições impostas.

O primeiro passo é realizado por um parser XML, pois a estrutura de cada uma das sintaxes de escrita de Topic Maps é definida formalmente. No caso de XTM, esta definição é feita pelo XTM 1.0 DTD (Pepper and Moore, 2001b).

Por sua vez, o processo de validação semântica de Topic Maps possui as seguintes etapas:

1. Percorrer o topic map concreto;
2. Para cada restrição, seleccionar os objectos adequados;
3. Validar os objectos seleccionados de acordo com a restrição;
4. Informar eventuais descumprimentos das regras em questão.

Ao que tudo indica, uma linguagem de especificação deve ser desenvolvida para descrever todos os tipos de restrições em Topic Maps.

A solução mais flexível é usar uma linguagem de especificação que permita descrever restrições em termos do modelo de dados da norma Topic Maps (Garshol et al., 2003). Isso significa que uma linguagem de programação está envolvida, o que vai implicar a criação de interfaces para que qualquer pessoa possa descrever restrições, e não somente os programadores.

A linguagem para a especificação de restrições também deverá permitir a definição de características de tópicos e associações. Assim, as restrições fornecerão uma clara indicação sobre o domínio tratado no topic map.

As restrições limitam a liberdade de criação de um topic map. No momento de determinar regras para um topic map, o projectista deve estar consciente dos limites que estas colocam; se elas são regras muito restritivas, podem reduzir consideravelmente a vantagem da liberdade de criação de topic maps, fornecida pela norma ISO 13250.

## 5.6 Sumário sobre Validação Semântica em Topic Maps

A conclusão a que se pretende chegar é que um sistema de restrições capaz de definir regras de validade semântica em topic maps é realmente necessário. Este sistema será útil no desenvolvimento de topic maps sobre qualquer domínio, para se ter a certeza de que o conhecimento descrito verifica as regras contextuais que lhe dão validade, isto é, que tornam o seu significado credível.

A partir das ideias apresentadas neste capítulo, decidiu-se pela criação de uma linguagem de especificação de restrições em Topic Maps baseada em XML. Assim, pode-se utilizar a linguagem XSLT para a criação de um processador para esta linguagem. Esta linguagem – e o respectivo processador – serão descritos no Capítulo 9, onde serão apresentados em detalhe casos onde as restrições podem ser aplicadas, demonstrando a sua aplicabilidade.

## Parte II

# Topic Maps: Um Ambiente Processamento



## Capítulo 6

# Metamorphosis

*Vem vamos embora  
que esperar não é saber  
quem sabe faz a hora  
não espera acontecer*

Geraldo Vandré

Diariamente, uma grande quantidade de dados é produzida em instituições. Para satisfazer os requisitos de armazenamento, estas organizações geralmente utilizam bases de dados relacionais, as quais são eficientes para guardar e manipular dados estruturados. Quando se trata de dados semi-estruturados, o armazenamento é realizado em documentos textuais ou anotados. Com isso, a integração da informação contida nessas fontes é uma tarefa árdua.

Diferentes abordagens podem ser seguidas para conseguir a integração, porém o problema principal será conquistar a interoperabilidade semântica entre as fontes de informação, garantindo a manipulação do conjunto sem forçar a conversão das partes em um formato único. As ontologias surgem como uma solução para este problema e, para a sua representação, os **Topic Maps** apresentam-se como uma alternativa atraente.

Esta é uma excelente situação real, actual, que ilustra a criação de *Topic Maps dirigidos por ontologias* (como dito no Capítulo 4) e as potencialidades da tecnologia **Topic Maps** para manusear informação (conforme discutido nos capítulos anteriores).

Neste contexto, este capítulo apresenta o **Metamorphosis** (Librelotto, Ramalho, and Henriques, 2004b) – um ambiente orientado a **Topic Maps** composto por módulos capazes de: extrair dados de recursos heterogêneos de informação, construir um topic map a partir da especificação da ontologia que responde aos requisitos de um determinado domínio; validar o topic map construído; e gerar um sistema de navegação conceptual que permite visualizar a informação como um todo, integrada e coerente.

## 6.1 Interoperabilidade em Sistemas de Informação

Para realmente se poder construir a tão falada Sociedade de Informação é necessário providenciar o acesso completo à informação disponível, a qual é geralmente heterogênea e distribuída. Para estabelecer uma partilha eficiente da informação, muitos problemas técnicos têm sido resolvidos. Primeiramente, uma fonte de informação apropriada deve ser encontrada, a qual contém dados relevantes para uma determinada tarefa. Encontrar recursos de informação apropriados é um problema tratado em áreas de recuperação de informação (*information retrieval*) e filtragem de informação (*information filtering*) (Wache et al., 2001).

Uma vez localizada a informação, é necessário assegurar o acesso às fontes de dados. Isto significa que os vários recursos de informação, encontrados no primeiro passo, devem poder ser manuseados uniformemente pelo sistema que, após a procura, vai fazer a extracção. O problema de fundir sistemas distribuídos e heterogêneos é conhecido como o *problema da interoperabilidade*.

Interoperabilidade é a habilidade de dois sistemas trocarem, interpretar e processarem correctamente a mesma informação (Lomet and Widom, 1995; Mitra and Wiederhold, 2001; Härder, Sauter, and Thomas, 1999; Albrecht, 1999). Isso requer algum grau de compatibilidade entre esses sistemas para que seja possível a troca de dados e sua correcta interpretação.

De uma forma geral, a partilha de informação não necessita somente fornecer acesso completo aos dados; ela também requer que os dados acedidos possam ser processados e interpretados pelos diversos sistemas remotos. Problemas que podem surgir da heterogeneidade dos dados são bem conhecidos na comunidade de sistemas de base de dados distribuídas (Kashyap and Sheth, 1996): heterogeneidade estrutural (heterogeneidade esquemática) e heterogeneidade semântica (heterogeneidade de dados) (Kim and Seo, 1991). Heterogeneidade estrutural significa que diferentes sistemas de informação armazenam seus dados em diferentes estruturas. Heterogeneidade semântica considera o conteúdo de um item de informação e seu significado pretendido.

Para conseguir interoperabilidade semântica em sistemas de informação heterogêneos, o significado da informação que é intercambiada deve ser compreendido por todos os sistemas (Newcomb, 2003). Conflitos semânticos ocorrem quando há uma ambiguidade na interpretação da informação. Goh (Goh, 1997) identifica três causas principais para heterogeneidade semântica:

- Conflitos ocorrem quando quando items de informação parecem ter o mesmo significado, mas de facto, diferem. Por exemplo, contextos temporais distintos;
- Conflitos de escala ocorrem quando diferentes sistemas de referência são usados para medir um valor. Exemplo disto são diferentes valores monetários;
- Conflitos de nomenclatura ocorrem quando os nomes em esquemas de informação diferem significativamente. Um fenómeno frequente é a presença de homónimos e sinónimos.

O ideal seria que todos os sistemas seguissem as mesmas normas computacionais (*standards*). Contudo, este nível de normalização é impossível de alcançar na prática, devido às taxas de mudanças tecnológicas, à falta de aceitação das normas internacionais, à existência de aplicações antigas não-compatíveis (*legacy systems*) ou, apenas, às razões de autonomia de cada sistema de informação.

De facto, uma aproximação cómoda, que se usou inicialmente mas que nos nossos dias é impraticável, é a fusão dos dados numa única fonte, o que implica converter a representação num formato único.

No lado oposto, estão as aproximações que não exigem conversão de formatos, mas que em vez disso, consegue-se a interoperabilidade através da publicação de interfaces, esquemas e formatos utilizados para o intercâmbio de informação, tornando sua semântica o mais explícita possível, para que ela possa ser devidamente manipulada pelos sistemas corporativos.

Hasselbring (Hasselbring, 2000) diz que a interoperabilidade de sistemas de informação deve ser considerada sob três pontos de vista: domínio da aplicação, projecto conceptual e tecnologia de sistemas. A Figura 6.1 ilustra a estrutura de um conjunto de informação e sua interoperabilidade em cada um desses pontos de vista.

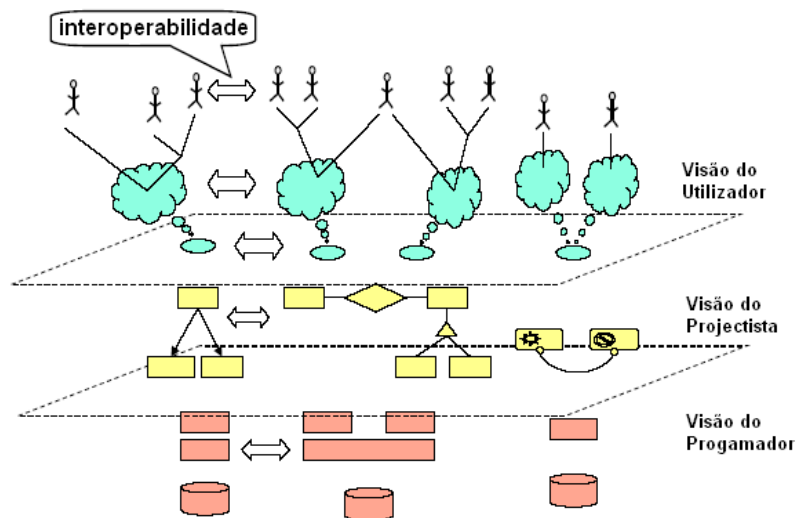


Figura 6.1: Perspectivas em interoperabilidade de sistemas de informação.

A *visão do utilizador* preocupa-se com as visões distintas sobre os recursos de informação e as especializações do domínio representado. A *visão do projectista* refere-se aos requisitos de modelação e projecto de sistemas. A *visão do programador* refere-se à tecnologia a usar para a implementação dos sistemas. Portanto, a interoperabilidade deve ser conquistada em todas estas camadas. Isso significa que:

- os utilizadores de um sistema devem entender a informação originada em outros sistemas;

- o projecto do sistema deve acomodar os dados importados;
- os programas devem automatizar o intercâmbio de informação (isto é, a transferência dos dados e suas transformações).

O problema mais difícil de interoperabilidade de dados ocorre na visão conceptual (do projectista) e de aplicação (do utilizador) (Abiteboul, Buneman, and Suciú, 1999). Visando fornecer uma solução para estes problemas, a próxima secção apresenta o **Metamorphosis**: um sistema que permite a integração de informação oriunda de diferentes fontes de dados, onde o conhecimento extraído é representado na norma ISO 13250 **Topic Maps**, sem alteração das fontes.

## 6.2 Introdução ao Metamorphosis

O paradigma tradicional para desenvolvimento de sistemas de informação é baseado no ciclo modelo-projecto-implementação, e considera um simples esquema utilizando um modelo de dados. O advento de sistemas heterogéneos e da Web está mudando este cenário. Grandes quantidades de dados estão disponíveis em formatos e plataformas distintas. Os repositórios de dados são muito variáveis e podem ir de sistemas de gestão de bases de dados a documentos estruturados. A falta de concordância na representação dos dados torna o problema da interoperabilidade muito complexo.

A possibilidade de conexões dinâmicas entre os componentes dos sistemas Web adiciona complexidade a este contexto. A demanda por interoperabilidade tem incentivado o desenvolvimento de normas e ferramentas para facilitar a transformação de dados e sua integração. No entanto, ainda há muitos desafios para serem solucionados, especialmente aqueles relacionados com a semântica dos dados. Um problema comum se encontra quando as organizações pretendem obter uma visão integrada de sistemas heterogéneos de informação, pois é necessário interrogar/extrair cada fonte de dados, sendo diferente o acesso a cada sistema.

Para resolver este problema, propõe-se o sistema **Metamorphosis** (Librelotto et al., 2004), um dos contributos principais deste trabalho de doutoramento, que torna possível a extração, validação, armazenamento e navegação em **Topic Maps**. Ele é composto por três módulos principais, conforme visualizado na Figura 6.2:

1. **Oveia**: consulta a meta-informação de sistemas heterogéneos de informação, e constrói – a partir da especificação de uma ontologia – um topic map, que é depois armazenado;
2. **XTche**: valida o topic map gerado, de acordo com um conjunto de restrições;
3. **Ulisses**: permite a navegação sobre o topic map, dando uma visão conceptual integrada sobre o domínio em questão e seus recursos.





Figura 6.2: Arquitectura simplificada do Metamorphosis

Deste modo, o Metamorphosis permite obter a interoperabilidade semântica em sistemas heterogéneos de informação porque os dados relevantes são extraídos e armazenados num topic map, de acordo com uma especificação da ontologia desejada. O ambiente valida o topic map gerado de acordo com um conjunto de regras definido numa linguagem para descrição de restrições. A navegação sobre o topic map proporciona uma visão homogénea sobre os recursos – o que justifica a decisão de chamar a este processo de *interoperabilidade semântica*.

### 6.3 Arquitectura do Metamorphosis

A principal ideia por trás do Metamorphosis é integrar a especificação de redes conceptuais expressadas em Topic Maps, com sua extracção automática e sua validação, assim como seu armazenamento e sua navegação. Para atingir este objectivo, o Metamorphosis toma como entrada:

**Recursos de Informação:** compostos por documentos XML, páginas HTML, bases de dados, etc. O Metamorphosis não modifica nenhuma fonte, apenas utiliza parte de seus dados para construir a rede semântica, através de uma ontologia definida para tal domínio;

**Especificações XML:** a descrição das fontes de dados (escrita em XSDS – *XML Specification for DataSources*); a descrição do topic map a ser construído (escrita em XS4TM – *XML Specification for Topic Maps*); e as condições de contexto, ou regras semânticas, a serem obedecidas por um topic map (escritas em XTche – *Topic Maps Schema and Constraint Language*).

e gera como saída:

**Um Website Conceptual:** O website gerado permite uma navegação através do sistema de informação dirigida por conceitos organizados de acordo com uma ontologia, representada num topic map, permitindo assim aceder aos recursos de informação.

A Figura 6.3 vem reforçar esta ideia e dá uma visão pictórica da arquitectura detalhada do Metamorphosis, que pode ser descrita da seguinte forma:

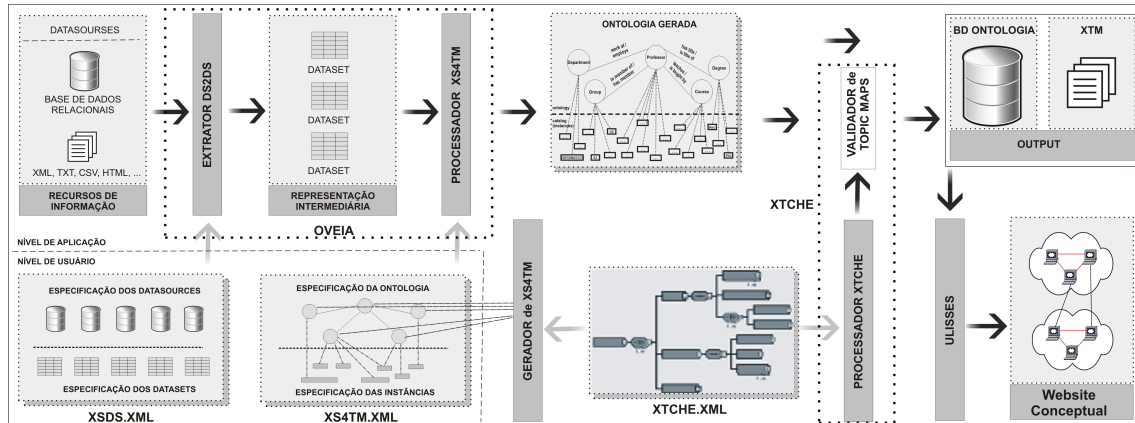


Figura 6.3: Arquitectura completa do Metamorphosis

- (1) **Camada de recursos de informação:** Composta pelas várias fontes de dados: documentos XML, bases de dados, páginas Web, etc.;
- (2) **Especificações XS4S e XS4TM :** São linguagens de domínio específico – escritas em formato XML – as quais definem o processo de construção de topic maps. XS4S especifica o que é necessário ser extraído das fontes de dados para a criação do topic map. XS4TM define como os dados extraídos formarão o topic map desejado. Estas linguagens são descritas no Capítulo 8;
- (3) **TM-Builder/Oveia:** O TM-Builder é um extractor de topic maps a partir de documentos XML (Capítulo 7). O Oveia, por sua vez, é um extractor de topic maps em sistemas heterogêneos de informação (Capítulo 8), sucedendo ao TM-Builder e resolvendo alguns dos problemas entretanto identificados – devido a este facto, o TM-Builder não é visualizado na Figura 6.3. Ambos os componentes utilizam uma linguagem de especificação de construção de topic maps para determinar como será montado o topic map a partir dos dados extraídos dos recursos de informação;
- (4) **Topic map gerado:** É o topic map obtido automaticamente, descrito na sintaxe XTM. Além da possibilidade de armazenamento em formato XTM, os topic maps criados pelo Oveia também podem ser armazenados em base de dados relacional. Para isso, a norma Topic Maps foi mapeada para um modelo relacional, chamado de BD Ontologia, conforme apresentado no Capítulo 8;
- (5) **Especificação XTche:** É uma linguagem de especificação de restrições para Topic Maps, baseada nos requisitos de TMCL (*Topic Map Constraint Language*) (Nishikawa, Moore, and Bogachev, 2004) definidos pela ISO/IEC. Permite especificar regras para a validação semântica de topic maps (Capítulo 9);
- (6) **Gerador de XS4TM:** A partir da especificação XTche, o Gerador de XS4TM efectua a criação do esqueleto da especificação XS4TM, criando os tipos de tópicos, tipos de associações e demais conceitos que foram definidos no esquema em XTche (Capítulo 9);

- (7) **Processador XTche:** Componente responsável pela validação de topic maps de acordo com um conjunto de restrições especificado na linguagem XTche (Capítulo 9). Se o topic map estiver de acordo com a especificação XTche, há uma confirmação da sua validade semântica; caso apresente alguma irregularidade, mensagens de erros serão mostradas, indicando os pontos em que o topic map não está correto;
- (8) **Ulisses:** Componente que toma como entrada um *topic map* e produz sua visualização na Web. O Ulisses tanto fornece a navegação conceptual a partir da sintaxe XTM, como a partir do modelo relacional BD Ontologia (Capítulo 10);
- (9) **Website:** É o website gerado através do qual é possível navegar semanticamente pelos vários recursos de informação que compõem o sistema de informação original.

Estes módulos – TM-Builder, Oveia, XTche e Ulisses – serão detalhados nos 4 capítulos seguintes desta dissertação.

Como se depreende do exposto, o funcionamento do Metamorphosis compõe-se de níveis, ou camadas independentes, sendo cada etapa realizada em cadeia, mas separada das restantes:

1. no Oveia, definem-se os ficheiros com as especificações das fontes de informação e da ontologia a ser usada para a extracção do topic map. Após o seu processamento, um sumário indica o tempo total e a quantidade de tópicos e associações extraídos para formar o topic map pretendido;
2. no XTche, indicam-se os nomes dos ficheiros que possuem: o topic map a ser validado; e o conjunto de restrições a ser aplicado. O resultado do processamento é a confirmação (ou não) da validade do topic map;
3. no Ulisses, indica-se qual o topic map que será processado para construir o website que oferece a almejada navegação semântica nos recursos de informação.

O funcionamento independente dos módulos permite que um componente em particular específica seja reimplementado, caso necessário, sem alterar os outros componentes pois o funcionamento de um módulo não interfere nos demais.

Porém, a característica mais importante é que a independência funcional e tecnológica<sup>1</sup> dos módulos permite usá-los isoladamente, reaproveitando as partes em contextos diferentes.

## 6.4 Aplicações do Metamorphosis

O Metamorphosis tem sido utilizado em vários projectos de pequena e média dimensão. Até ao momento, provou ser uma boa ferramenta de prototipagem. As interfaces Web são criadas rapidamente e sem grandes dificuldades por parte dos utilizadores.

Desta forma, o Metamorphosis pode ser aplicado nas seguintes áreas de intervenção:

---

<sup>1</sup>Cada módulo está implementado com arquitecturas e técnicas de programação distintas.

**Conferências e Workshops** – o sistema tem sido utilizado para suportar os websites de algumas conferências. O uso do Metamorphosis para a criação do website para a conferência portuguesa intitulada XATA (*XML, Aplicações e Tecnologias Associadas*) foi apresentado em (Librelotto, Ramalho, and Henriques, 2003b);

**E-Learning** - o Metamorphosis é usado por uma plataforma para a produção de conteúdos baseada na tecnologia XML, chamada Adrian (Librelotto, Ramalho, and Henriques, 2004a). Desta iniciativa, surgiram pequenos projectos que deram origem a várias aplicações XML: uma para a produção de guiões para aulas, outra para a produção de apresentações e outra para a produção de testes e exames (Librelotto, Ramalho, and Henriques, 2003a). Para integrar todas as aplicações, está-se a utilizar o Metamorphosis para a criação da interface Web;

**Arquivos Históricos** - Estão em curso vários projectos, em dois arquivos históricos portugueses, que visam a disponibilização do acervo documental daqueles arquivos na Web. O sistema de informação por detrás de um arquivo histórico é gigantesco e o Metamorphosis está a ser utilizado para prototipar a interface Web. O Capítulo 14 apresenta um caso de estudo neste contexto.

**Interfaces Web instantâneas** - Muitas vezes, para se testar algumas funcionalidades de um sistema que se está a desenvolver é necessário criar uma interface Web para realizar esses testes. Normalmente, estas interfaces não têm grandes requisitos quanto à aparência, o que se pretende é que sejam desenvolvidas o mais rapidamente possível.

**Geração de índices para um recurso de informação** - Quer-se expor na Web um sistema de informação composto por bases de dados, documentos XML, documentos PDF, etc. Quer-se que toda a informação esteja acessível via Web e para isso constroem-se, manualmente ou automaticamente, os primeiros índices. Estes índices acabam por ser enormes, excedendo a capacidade dos browsers actuais. Pode-se fraccioná-los alfabeticamente, mas há situações em que isso não é possível, nem recomendável. Contudo, é possível arranjar um método para fraccionar a informação conceptualmente. O Metamorphosis então pode ser utilizado para a criação de um topic map, o qual pode ser visto também como uma indexação dos recursos.

## 6.5 Sumário sobre o Metamorphosis

Este capítulo introduziu um ambiente que proporciona a integração de sistemas heterogéneos de informação usando a norma **Topic Maps**, gerando uma visão homogénea destes recursos. A proposta – que será detalhada nos restantes capítulos da dissertação (explicação dos módulos e exploração da plataforma) – é um ambiente, chamado **Metamorphosis**, para a construção automática de topic maps com dados extraídos de várias fontes e a disponibilização de uma navegação conceptual sobre a informação extraída.

Embora o Metamorphosis tenha sido desenvolvido para uso nas nossas áreas de trabalho principais – processamento de documentos XML aplicado a Arquivos Públicos e Museus Virtuais – estamos convencidos que o Metamorphosis pode ser aplicado com sucesso na

área geral de sistemas de informação para a integração de dados, análise e exploração de conhecimento. O **Metamorphosis** tem sido utilizado em vários projectos de pequena e média dimensão. As interfaces Web são criadas rapidamente e sem grandes dificuldades – em termos de especificação – por parte dos utilizadores.

Em termos de trabalhos relacionados, até o presente momento não é do conhecimento dos autores nenhuma ferramenta com as mesmas características do **Metamorphosis**. Contudo, pode-se comparar, individualmente, cada uma das suas partes com ferramentas conhecidas, o que será feito nos capítulos referentes a cada módulo do **Metamorphosis**:

- Uma comparação entre o **Oveia**, o **TSIMMIS** (Garcia-Molina et al., 1997) e o **KAON REVERSE** (Handschuh et al., 2001) é apresentada no Capítulo 8;
- O **XTche** é comparado no Capítulo 9 com o *AsTMa!* (Barta, 2003a) e com uma proposta de Eric Freese (Freese, 2002);
- Uma comparação entre o **Ulisses** e o **Ontopia Omnigator** (Ontopia, 2002b) é encontrada no Capítulo 10;



## Capítulo 7

# TM-Builder

*Não deixe de fazer algo que gosta devido a falta de tempo,  
a única falta que terá,  
será desse tempo que não volta mais.*

Mário Quintana

Ao longo desta dissertação tem-se vindo a defender o uso de **Topic Maps** dirigidos por ontologia para descrever (e acender) o conhecimento armazenado em fontes de informação diversos.

Essa ideia foi colmatada, no capítulo anterior, com a proposta de um ambiente integrado para a criação, validação e navegação dos ditos **Topic Maps** – o **Metamorphosis**. Para explicar este sistema, é agora necessário descrever o seu primeiro patamar, dizendo como se pode construir os **Topic Maps** a partir das referidas fontes.

Note-se que a ideia é, não só mecanizar a construção de **Topic Maps** à custa de uma ferramenta de extracção, como também automatizar a criação dessa ferramenta.

Assim, este capítulo tem como objectivo apresentar um gerador de extractores de **Topic Maps** a partir de documentos XML. A criação destes extractores – denominados **TM-Builder** – é definida através de uma linguagem de domínio específico chamada **XSTM** (*XML Specification for Topic Maps*), a qual descreve o processo de extracção de dados, a partir de elementos e atributos de documentos XML, para a criação de um topic map (que será representado na sintaxe XTM).

O **TM-Builder** baseou-se em uma arquitectura abstracta de um gerador de construtores de ontologias (**Ontology Builder**) a partir de uma família de documentos XML. O **Ontology Builder** (**OntBuild**) seria obtido através de uma linguagem de especificação de extracção de ontologias, a qual deu origem à linguagem **XSTM**.

Para apresentar e defender as ideias acima citadas, este capítulo está assim organizado: na Secção 7.1, será apresentado o extractor genérico de ontologias – **OntBuild** – que é o modelo defendido; o extractor de **Topic Maps** a partir de documentos XML – **TM-Builder** –

é descrito na Secção 7.2; a definição da linguagem XSTM será encontrada na Secção 7.3; os trabalhos relacionados com o TM-Builder são citados na Secção 7.4; por fim, um sumário deste capítulo é apresentado na Secção 7.5.

## 7.1 OntBuild: O Extractor Abstracto de Ontologias

Após a criação manual de algumas ontologias, descritas em diferentes sintaxes XML conhecidas (citadas na Secção 2.4), verificou-se que esta tarefa, além de consumir tempo, é bastante complexa.

Por mais que existam ferramentas que facilitam o processo, como o *Protégé 2000* (Protégé, 2005) – permitindo uma especificação gráfica de ontologias – um problema ainda deve ser solucionado: o aproveitamento da informação relevante para sua construção que está contida em fontes de dados (por exemplo, documentos XML). O ideal seria que a informação armazenada pudesse ser obtida de forma automática, evitando assim uma edição individual de cada item da ontologia.

A fim de facilitar a criação de uma ontologia a partir de dados contidos em documentos XML, decidiu-se pela definição de um extractor que tivesse a capacidade de a construir automaticamente, com base numa especificação que explicita quais os elementos e atributos de tais documentos XML devem ser retirados, e como estes devem ser associados. A esse extractor, foi dado o nome *Ontology Builder*, ou simplesmente **OntBuild**.

Contudo, mesmo recorrendo aos serviços de um extractor, como o **OntBuild**, a tarefa de criar uma ontologia, para conter a representação do conhecimento de um domínio pretendido, é complexa e lenta: isso porque é necessário conhecer ao pormenor a estrutura (o DTD ou XML Schema) da família de documentos a processar de modo a explicitar todos os elementos e atributos que se pretende extrair e relacionar. Além disso, o **OntBuild** é fortemente dependente da estrutura dos recursos de informação; trabalhando com várias famílias de documentos, é necessário implementar extractores **OntBuild** distintos, sendo um para cada esquema de documento diferente.

Para minimizar o esforço envolvido, foi estabelecida uma linguagem de domínio específico para derivação sistemática de um extractor de ontologia a partir de documentos XML.

Para poder concretizar a ideia de gerar o **OntBuild**, é necessário especificar formalmente o processo de extracção para cada tipo de documentos a processar. Esta linguagem abstracta para especificação de ontologias, chamada XSO (*XML Specification for Ontologies*), permite a especificação das partes de informação que devem ser extraídas, assim como determina a maneira como estes dados devem estar relacionados entre si, criando uma rede semântica de acordo com a ideia do projectista. A Figura 7.1 ilustra o sistema completo que se pretende construir.

Na prática, após o processamento da especificação XSO pelo processador XSO-P, um extractor de ontologias (**OntBuild**) é gerado. O extractor obtido é específico para a extracção de ontologia numa família de documentos.

O processador XSO (XSO-P) é o gerador automático de **OntBuilds**. A partir de uma es-



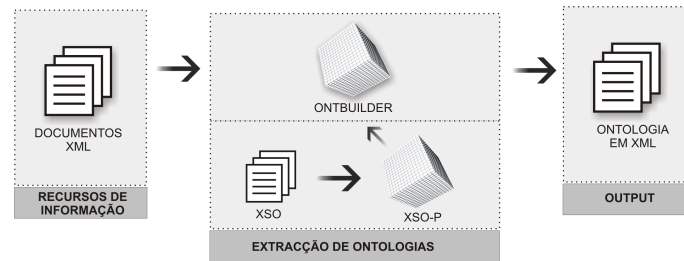


Figura 7.1: A arquitectura do sistema OntBuild

pecificação em XSO, tal processador gera um OntBuild, o qual que irá processar o conjunto de documentos de entrada para extrair a ontologia desejada.

### 7.1.1 Uma linguagem para representação de ontologias no OntBuild

Como citado anteriormente, o OntBuild é uma arquitectura abstracta e genérica para a criação de extractores de ontologias. Para sua efectiva implementação é necessário definir qual das principais normas para representação de ontologias baseada em XML será adoptada pelo OntBuild.

Para esta decisão, foram analisadas as características de cada uma das alternativas mais importantes apresentadas para descrição de ontologias descritas na Secção 2.4: RDF (*Resource Description Framework*) (Lassila and Swick, 1999), XOL (*XML-based Ontology Exchange Language*) (Karp, Chaudhri, and Thomere, 1999), SHOE (*Simple HTML Ontology Extensions*) (Luke and Helfin, 2000), OWL (*Web Ontology Language*) (Bechhofer et al., 2002) e Topic Maps (Pepper and Moore, 2001a).

Na Subsecção 3.5.5 foram apresentados vários argumentos que nos levaram à escolha da norma Topic Maps para a representação de conhecimento e à sintaxe XTM (*XML Topic Maps*). A arquitectura abstracta OntBuild foi, então, adaptada para permitir a geração de topic maps em XTM, como formato de saída. Sendo assim, a linguagem XSO passa a ser denominada por linguagem XSTM (*XML Specification for Topic Maps*) e o extractor – genericamente designado OntBuild – passa a denominar-se TM-Builder – *Topic Maps Builder*.

## 7.2 TM-Builder: Um extractor de Topic Maps

Para poder concretizar esta ideia de gerar o TM-Builder, é necessário especificar formalmente o processo de extracção concreto, para cada tipo de documentos a processar. Como se disse acima, esta especificação será feita em XSTM.

O processador de XSTM (XSTM-P) é o gerador de TM-Builders e é uma folha de estilos XSL. A partir de uma especificação em XSTM, tal processador (que vulgarmente é

chamado XSL de segundo nível<sup>1</sup>) gera um TM-Builder que irá processar o conjunto de documentos de entrada para extrair o topic map e representá-lo na sintaxe XTM.

Em termos práticos, um TM-Builder é também uma folha de estilos XSL que recebe um documento XML (ou um conjunto de documentos de um mesmo tipo) e gera um topic map representado na sintaxe XTM.

Esta abordagem oferece ao utilizador o ambiente de trabalho, completamente XML, cuja arquitectura genérica é ilustrada na Figura 7.2. É claro que essa arquitectura é uma especialização, ou concretização, da arquitectura abstracta apresentada na Figura 7.1.

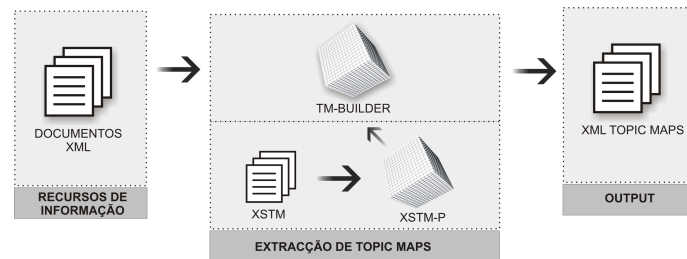


Figura 7.2: Arquitectura do TM-Builder

Ambas as folhas de estilo XSL (o gerador de extractor e o próprio extractor) são processados por um processador XSL standard, como Saxon<sup>2</sup> ou Xalan<sup>3</sup>, sendo este mais um dos benefícios desta proposta.

### 7.3 XSTM: Uma linguagem XML para especificar Extractores de Topic Maps

A linguagem XSTM (Librelotto, Ramalho, and Henriques, 2003d) é um dialecto XML que permite especificar o processo de extracção do topic map que se pretende construir ao analisar documentos anotados pertencentes a um mesmo esquema XML.

A linguagem XSTM fornece os principais construtores para a criação de Topic Maps – seja para a definição da ontologia, seja para a base de conhecimento associada a ela.

De um modo formal, a *Gramática Independente de Contexto* (GIC) da linguagem XSTM é mostrada abaixo:

```

1 | XSTM      ::= TopicType+ Topic+ AssocType* Assoc*
2 | Topic    ::= xpath TTypeID
3 | TopicType ::= tTypeID tTypeName
4 | Assoc    ::= AssocClass aTypeID LElem RElem
5 | AssocClass ::= one2one || m2n
6 | AssocType ::= aTypeID aTypeName LElem RElem
7 | LElem    ::= tTypeID eName? tTypeID
8 | RElem    ::= tTypeID eName? param? tTypeID

```

<sup>1</sup>XSL de segundo nível significa uma folha de estilos XSL que gera outra folha de estilos XSL.

<sup>2</sup><http://saxon.sourceforge.net/>

<sup>3</sup><http://xml.apache.org/xalan-j/>

A partir da definição formal apresentado na GIC acima, mapeou-se o modelo formal da linguagem XSTM para XML. Com isso, obteve-se um DTD (*Document Type Definition*) e um XML-Schema que descrevem a gramática de XSTM, de modo a permitir o seu uso em todos os ambientes de processamento XML.

O esquema básico da linguagem XSTM pode ser visualizado na Figura 7.3, onde percebe-se que seu elemento raiz é `<xstm>`, o qual é composto pela sequência da definição:

- dos tipos de tópico (elemento *topicType*);
- da extracção dos tópicos (elemento `<topic>`);
- dos tipos de associações (elemento *assocType*);
- da extracção das associações (elemento *assoc*).

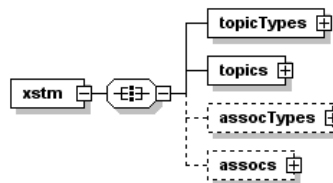


Figura 7.3: Esquema do elemento raiz de XSTM

Como em *Topic Maps* tudo pode ser visto como tópicos e associações, XSTM resume-se à definição de seus elementos justamente nestes termos, além de seus tipos. A semântica de cada tópico é definida em sua especificação em particular.

O esquema de XSTM será completado com o esquema de cada um dos elementos acima apresentados, o que é realizado em cada uma das subsecções posteriores.

### 7.3.1 Especificação de Tipos de Tópicos em XSTM

Os tipos de tópicos são declarados no elemento `<topicTypes>`, cuja estrutura é formada por dois sub-elementos: `<firstTopicType>` e `<topicType>`.

O elemento `<topicType>` contém três sub-elementos: um identificador – `<id>` – para ser referenciado pelas suas instâncias; um identificador para o tipo mais genérico que este especializa (sua super-classe) – `<instanceOf>`; e um nome para a visualização do tipo de tópico em um navegador – `<name>`. Essa estrutura é apresentada na Figura 7.4.

Por sua vez, o elemento `<firstTopicType>` define qual tipo de tópico será considerado o raiz da árvore hierárquica do topic map. Ele possui a mesma estrutura do elemento `<topicType>`. Porém como este é o raiz da hierarquia de tópicos, obviamente não possui um tipo-pai. Sendo assim, o sub-elemento `<instanceOf>` não se faz necessário. O nome deste tipo de tópico será apresentado sempre no princípio da navegação/visualização deste topic map.

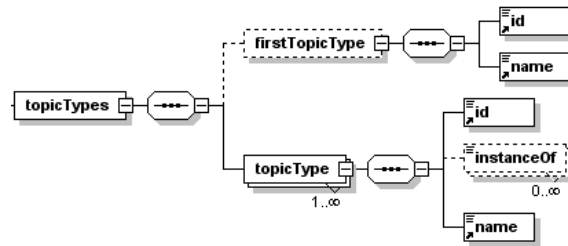


Figura 7.4: Estrutura de um tipo de tópico em XSTM

A fim de demonstrar o uso do TM-Builder para a extração de ontologias a partir de uma fonte XML, usar-se-á o caso de estudo da workshop *XML, Aplicações e Tecnologias Associadas* (XATA<sup>4</sup>).

O XML-Schema dos documentos XML que contém a informação sobre esse evento é apresentado na Figura 7.5. Obviamente este XML-Schema está incompleto, porém, para esse caso de estudo, o importante está ressaltado nessa figura.

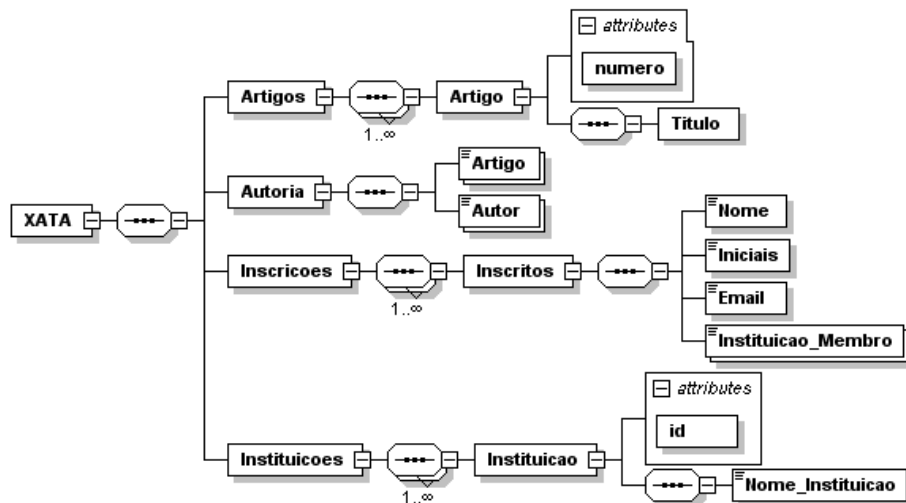


Figura 7.5: XML-Schema do XATA

O exemplo a seguir apresenta um trecho da especificação XSTM para o XATA, identificando o tópico raiz da hierarquia de classes do topic map, assim como três de suas instâncias: *autor*, *artigo* e *instituição*.

```

1 <xstm>
2   <topicTypes>
3     <firstTopicType>
4       <id>XATA</id>
5       <name>XATA: XML, Aplicações e Tecnologias Associadas</name>
6     </firstTopicType>
7     <topicType>
8       <id>autor</id>

```

<sup>4</sup>XATA: XML, Aplicações e Tecnologias Associadas – <http://vecpar.fe.up.pt/xata2005/>

```

9      <instanceOf>XATA</instanceOf>
10     <name>Autor de Artigo</name>
11  </topicType>
12  <topicType>
13     <id>artigo</id>
14     <instanceOf>XATA</instanceOf>
15     <name>Artigo</name>
16  </topicType>
17  <topicType>
18     <id>instituicao</id>
19     <instanceOf>XATA</instanceOf>
20     <name>Instituição</name>
21  </topicType>
22  ...
23 </topicTypes>
24 ...
25 </xstm>

```

O exemplo acima define o tópicos *XATA* como raiz, o qual possui as instâncias *autor*, *artigo* e *instituição*. O código XTM abaixo representa os tópicos que serão gerados pelo TM-Builder a partir dessa especificação:

```

1  <topic id="id_xata">
2    <baseName>
3      <baseNameString>XATA: XML, Aplicações e Tecnologias Associadas</baseNameString>
4    </baseName>
5  </topic>
6  <topic id="id_autor">
7    <instanceOf>
8      <topicRef xlink:href="#id_xata"/>
9    </instanceOf>
10   <baseName>
11     <baseNameString>Autor de Artigo</baseNameString>
12   </baseName>
13 </topic>
14 <topic id="id_artigo">
15   <instanceOf>
16     <topicRef xlink:href="#id_xata"/>
17   </instanceOf>
18   <baseName>
19     <baseNameString>Artigo</baseNameString>
20   </baseName>
21 </topic>
22 <topic id="id_instituicao">
23   <instanceOf>
24     <topicRef xlink:href="#id_xata"/>
25   </instanceOf>
26   <baseName>
27     <baseNameString>Instituição</baseNameString>
28   </baseName>
29 </topic>

```

Nos tópicos gerados percebe-se que o conteúdo dos identificadores (atributos @id) e das referências (atributos @xlink:href) sempre se iniciam com o texto “id\_”. Isso é feito automaticamente pelo TM-Builder para evitar as situações onde o conteúdo do elemento que forma o identificador inicia-se com um caracter diferente de literal. Além disso, todos os caracteres que não devem ser encontrados em um identificador em XML são eliminados.

Portanto, para cada tipo de tópico definido no elemento `<topicType>` em XSTM, será criado um tópico no topic map gerado pelo TM-Builder. Portanto, um mapeamento 1x1.

### 7.3.2 Especificação de extracção de Tópicos em XSTM

Enquanto os tipos de tópicos são conceitos abstractos definidos pelo projectista da representação do domínio, os tópicos são extraídos pelo TM-Builder a partir dos documentos XML tomados como entrada. Para a definição deste processo de extracção usa-se o elemento `<topics>`.

Este elemento é formado por uma sequência de vários sub-elementos `<topic>`, cuja estrutura encontra-se apresentada na Figura 7.6 e é composta pelos seguintes sub-elementos:

- `<xpath>` - é o caminho XPath referente ao elemento desejado, incluindo o caminho para os elementos que serão utilizados como identificador (`@id`) e como nome dos tópicos extraídos (`@name`);
- `<instanceOf>` - o tipo deste tópico (referência ao identificador de um tipo de tópico previamente definido);
- `<resourceData>` - o caminho XPath para um elemento que contenha algum dado, em forma textual, que forneça informação sobre o tópico. Por exemplo: a idade de uma pessoa, a morada de uma empresa, o número de um passaporte, etc. É um elemento não obrigatório em XSTM;
- `<resourceRef>` - o caminho XPath para um elemento que faça referência a algum recurso externo (uma URI) que descreva alguma característica do tópico. Por exemplo: o website de uma instituição, o e-mail de uma pessoa, etc. Também é um elemento não obrigatório no esquema de XSTM.

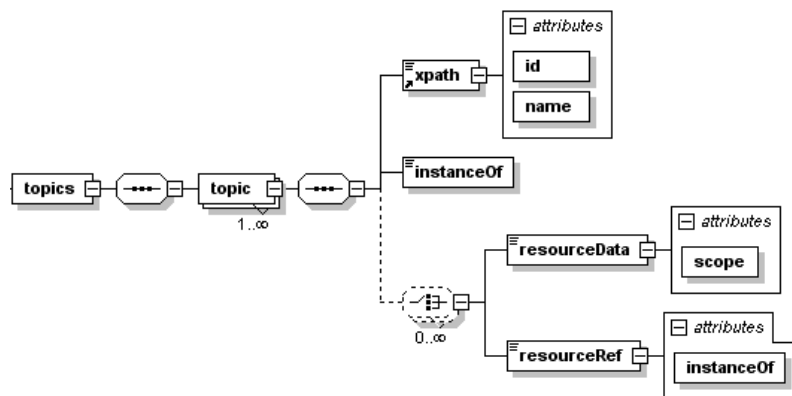


Figura 7.6: Estrutura da definição de extracção de tópicos em XSTM

Para exemplificar, o segmento de código XSTM abaixo (também retirado do Caso de Estudo do XATA) apresenta a definição da extracção dos tópicos que são instâncias dos tipos de tópico *autor*, *artigo* e *instituição*.

```

1 <xstm>
2   <topics>
3     <topic>
4       <xpath id="Iniciais" name="Nome">//Inscricoes/Inscritos</xpath>
5       <instanceOf>autor</instanceOf>
6       <resourceRef instanceOf="e-mail">Email</resourceRef>
7     </topic>
8     <topic>
9       <xpath id="@numero" name="Titulo">//Artigos/Artigo</xpath>
10      <instanceOf>artigo</instanceOf>
11    </topic>
12    <topic>
13      <xpath id="@id" name="Nome_Instituicao">//Instituicoes/Instituicao</xpath>
14      <instanceOf>instituicao</instanceOf>
15    </topic>
16  </topics>
17 </xstm>

```

Tomando o exemplo da definição da extração dos tópicos instâncias de *autor*, percebe-se que:

- cada tópico deste tipo será extraído dos elementos `<Inscritos>` dos documentos fontes que sejam acessíveis através do caminho XPath `//Inscricoes/Inscritos`;
- o identificador de cada tópico deste tipo será obtido no caminho XPath `//Inscricoes/Inscritos/Iniciais`;
- seu nome será obtido no caminho XPath `//Inscricoes/Inscritos/Nome`;
- uma referência a um recurso externo, do tipo *e-mail*, será extraído do caminho XPath `//Inscricoes/Inscritos/Email`.

Desta forma, o autor abaixo apresentado em XML...

```

1 <Inscricoes>
2   <Inscritos>
3     <Nome>José Carlos Ramalho</Nome>
4     <Iniciais>jcr</Iniciais>
5     <Email>jcr@di.uminho.pt</Email>
6     <Instituicao_Membro>UMinho</Instituicao_Membro>
7   </Inscritos>
8 </Inscricoes>

```

irá determinar a criação do seguinte tópico:

```

1 <topic id="id_jcr">
2   <instanceOf>
3     <topicRef xlink:href="#id_autor"/>
4   </instanceOf>
5   <baseName>
6     <baseNameString>José Carlos Ramalho</baseNameString>
7   </baseName>
8   <occurrence>
9     <instanceOf>
10      <topicRef xlink:href="#id_email"/>

```

```

11         </instanceOf>
12         <resourceRef xlink:href="mailto:jcr@di.uminho.pt"/>
13     </occurrence>
14 </topic>

```

Este processo de extracção irá ser realizado para todos os elementos encontrados no caminho XPath `//Inscricoes/Inscritos`. Assim sendo, uma única especificação XSTM define a extracção de todos os tópicos de cada tipo, independente da quantidade dos mesmos na fonte XML; portanto, um mapeamento 1xN.

### 7.3.3 Especificação de Tipos de Associações em XSTM

Uma vez definidos os tópicos e seus tipos – e antes de especificar a extracção das associações – o próximo passo é a definição dos tipos de associação. Os tipos de associação fazem parte dos conceitos abstractos de *Topic Maps* (assim como os tipos de tópicos) definindo os papéis de actuação e os nomes destes papéis em cada associação.

Em XSTM, os tipos de associação são declarados com o elemento `<assocType>`, o qual possui um identificador – `<id>` – e um nome para caracterizar este tipo – `<name>`. A Figura 7.7 apresenta o seu esquema.

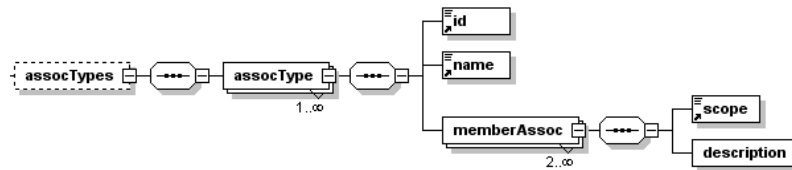


Figura 7.7: Estrutura de um tipo de associação em XSTM

Além disso, define-se os papéis de actuação deste tipo de associação, no elemento `<memberAssoc>`. Cada membro é definido por um contexto (elemento `scope`) – o identificador deste papel – e a sua respectiva descrição textual (elemento `description`). Cada um dos papéis de actuação declarados no elemento `<memberAssoc>` define a formação de um novo tópico, no topic map resultante.

Como exemplo, tem-se a especificação do tipo de associação que envolve os tópicos *autor* e *instituição*:

```

1 <xstm>
2   <assocTypes>
3     <assocType>
4       <id>autor_instituicao</id>
5       <name>Associação entre Autor e Instituição</name>
6       <memberAssoc>
7         <scope>representa</scope>
8         <description>é representada pelo autor</description>
9       </memberAssoc>
10      <memberAssoc>
11        <scope>pertence</scope>
12        <description>pertence a instituição</description>
13      </memberAssoc>

```



```

14         </assocType>
15         ...
16     </assocTypes>
17 </xstm>

```

A declaração XSTM acima define a criação dos seguintes tópicos, os quais estão representados em XTM logo a seguir:

**Tópico id\_autor\_instituicao:** é o tipo de associação que envolve autores e instituições (linhas 1 a 17);

**Tópico id\_representa:** é um dos papéis de associação (linha 18);

**Tópico id\_pertence:** é outro dos papéis de associação (linha 19).

```

1  <topic id="id_autor_artigo">
2    <baseName>
3      <baseNameString>Associação entre Autor e Artigo</baseNameString>
4    </baseName>
5    <baseName>
6      <scope>
7        <topicRef xlink:href="#id_representa"/>
8      </scope>
9      <baseNameString>é representada pelo autor</baseNameString>
10   </baseName>
11   <baseName>
12     <scope>
13       <topicRef xlink:href="#id_pertence"/>
14     </scope>
15     <baseNameString>escreve o artigo</baseNameString>
16   </baseName>
17 </topic>
18 <topic id="id_representa"/>
19 <topic id="id_pertence"/>

```

Cada definição de um tipo de associação em XSTM vai gerar três tópicos no topic map resultante: um referente ao tipo de associação em questão e outros dois referindo os papéis de actuação presentes neste tipo de associação. Representa, portanto, um mapeamento 1x3.

#### 7.3.4 Especificação de extracção de Associações em XSTM

Para finalizar a especificação XSTM, o elemento `<assocs>` permite a especificação de todas as associações a serem extraídas a partir do documento XML fonte. A estrutura deste elemento está apresentada na Figura 7.8.

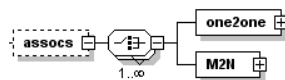


Figura 7.8: Estrutura da definição de extracção de associações em XSTM

XSTM permite a definição de três tipos de relacionamentos entre elementos e atributos XML, os quais são descritos por dois sub-elementos alternativos de `<assocs>`:

- o elemento `<one2one>` descreve as associações entre elementos que possuam sub-elementos (ou atributos) com conteúdos idênticos;
- o elemento `<one2one>` descreve as associações entre elementos que não possuam sub-elementos (ou atributos) com conteúdos idênticos; neste caso, um outro elemento define a relação através de alguma referência.

### Associações *one2one*

A estrutura dos elementos `<one2one>` é composta por uma referência ao seu tipo de associação – `<instanceOf>` – e a definição dos membros que fazem parte desta associação – `<members/element>`. Essa estrutura apresenta-se na Figura 7.9.

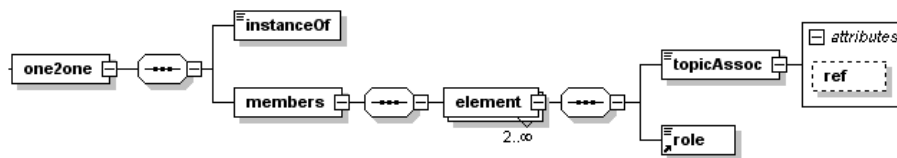


Figura 7.9: Estrutura da definição de extracção de associações *one2one* em XSTM

O caminho XPath `<members/element>` é composto por dois sub-elementos:

1. o elemento `<topicAssoc>` define o tipo de tópico do membro pertencente a esta associação;
2. o elemento `<role>`, que identifica o papel de actuação deste membro na actual associação.

O caminho XPath `//topicAssoc/@ref` contém o caminho XPath para o sub-elemento (ou atributo) do tópico membro em questão que deve ter o conteúdo idêntico à referência do outro membro da associação; ou seja, uma associação será criada quando o conteúdo das expressões XPath<sup>5</sup> dos tópicos pertencentes a essa associação forem iguais.

Com fins ilustrativos, pode-se definir uma associação entre *autor* e *instituição* a partir do esquema do XATA apresentado na Figura 7.5, pois percebe-se que há uma referência à instituição de cada autor no caminho XPath `//Inscrito/Instituicao_Membro`.

O trecho abaixo, do documento XML referente ao XATA, mostra que a instituição representada pelo membro *José Carlos Ramalho* (definido na Subsecção 7.3.2) – identificada por *UMinho* – é a Universidade do Minho:

<sup>5</sup> Expressões XPath contidas nos caminhos `//topicAssoc/@ref` de uma mesma associação M2N.

```

1 | <Instituicoes>
2 |   <Instituicao id="UMinho">
3 |     <Nome_Instituicao>Departamento de Informática - Universidade do Minho</Nome_Instituicao>
4 |   </Instituicao>
5 |   ...
6 | </Instituicoes>

```

Sendo assim, o seguinte código XSTM deve ser escrito para definir as associações entre autores e instituições:

```

1 | <assocs>
2 |   <one2one>
3 |     <instanceOf>autor_instituicao</instanceOf>
4 |     <members>
5 |       <element>
6 |         <topicAssoc ref="@id">instituicao</topicAssoc>
7 |         <role>representada</role>
8 |       </element>
9 |       <element>
10 |        <topicAssoc ref="Instituicao_Membro">autor</topicAssoc>
11 |        <role>pertence</role>
12 |      </element>
13 |    </members>
14 |  </one2one>
15 | </assocs>

```

A especificação de extracção de associações acima apresentada define que para todo elemento `//Inscricoes/Inscritos/Instituicao_Membro` será procurado um elemento no caminho XPath `//Instituicoes/Instituicao/@id` com o conteúdo idêntico. Quando isso ocorrer – como foi no caso de *José Carlos Ramalho* e a instituição *Universidade do Minho*, pois o conteúdo do caminho XPath em ambos os casos é *UMinho* – será criada uma nova associação no topic map.

Os caminhos XPath acima citados foram formados a partir da união do conteúdo do caminho XPath em XSTM de cada tipo de tópico (`//topic/xpath`) com o elemento que define a associação (`//members/element/topicAssoc/@ref`). A Tabela 7.1 apresenta o caso entre *autores* e *instituições*.

Tabela 7.1: Associação *one2one* entre Autor e Instituição

Tópico	Autores	Instituição
(1) <code>//topic/xpath</code>	<code>//Inscricoes/Inscritos</code>	<code>//Instituicoes/Instituicao</code>
(2) <code>//members/element/topicAssoc/@ref</code>	<code>Instituicao_Membro</code>	<code>@id</code>
Concatenação (1)+(2)	<code>//Inscricoes/Inscritos/Instituicao_Membro</code>	<code>//Instituicoes/Instituicao/@id</code>

A associação criada por esta especificação está abaixo representada em XTM:

```

1 | <association id="id_jcr_uminho">
2 |   <instanceOf>
3 |     <topicRef xlink:href="#id_autor_instituicao"/>
4 |   </instanceOf>
5 |   <member>
6 |     <roleSpec>
7 |       <topicRef xlink:href="#id_pertence"/>
8 |     </roleSpec>
9 |     <topicRef xlink:href="#id_jcr"/>

```

```

10 |         </member>
11 |     <member>
12 |         <roleSpec>
13 |             <topicRef xlink:href="#id_representada"/>
14 |         </roleSpec>
15 |         <topicRef xlink:href="#id_uminho"/>
16 |     </member>
17 | </association>

```

O tipo de associação e os papéis de actuação em todas as associações criadas serão idênticos; o que altera de uma associação para outra são os tópicos membros, os quais são extraídos do XML fonte. Assim, o mapeamento é como o da especificação de extracção de tópicos: 1xN (1 especificação – N associações geradas).

### Associação *M2N*

Quando não há elementos com o conteúdo idêntico em ambos os tópicos envolvidos na associação, mas há um elemento de referência, recomenda-se o uso da declaração *M2N*.

Esta situação geralmente ocorre quando se descarrega o conteúdo de uma base de dados para um documento XML. Quando ocorre uma relação *muitos-para-muitos*, no modelo relacional, define-se uma tabela de referência que armazenará as chaves primárias das tabelas envolvidas no relacionamento em questão.

Um exemplo similar é mostrado na Figura 7.5, onde o elemento <Autoria> associa os autores (elemento <Inscritos>) e os artigos (elemento <Artigos>). Um trecho do documento XML do XATA está abaixo representado:

```

1 | <Artigo numero="a15">
2 |     <Titulo>XML Templates for Constraints (XTC), Um Nível de Abstracção para Linguagens de
3 |         Especificação de Restrições</Titulo>
4 | </Artigo>
5 | <Autoria>
6 |     <Artigo>a15</Artigo>
7 |     <Autor>jcr</Autor>
8 | </Autoria>

```

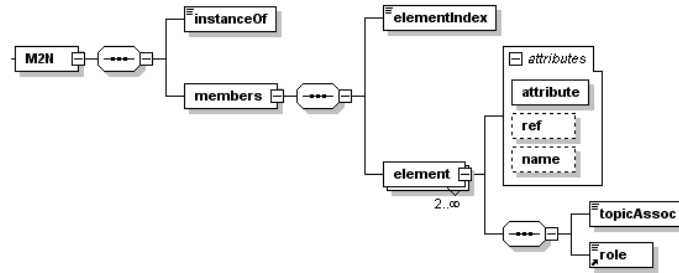
Esse trecho de documento XML indica que o autor identificado por *jcr*, cujo nome é *José Carlos Ramalho* (conforme XML da Subsecção 7.3.2), é autor do artigo identificado por *a15* que possui o título *XML Templates for Constraints (XTC), Um Nível de Abstracção para Linguagens de Especificação de Restrições*.

Quando houver a necessidade de associar os artigos com os seus autores, precisa-se referenciar a tabela *Autoria*, a qual determina esta associação. A especificação XSTM, definida pela estrutura descrita na Figura 7.10, seria então escrita assim:

```

1 | <M2N>
2 |     <instanceOf>autor_artigo</instanceOf>
3 |     <members>
4 |         <elementIndex>//Autoria</elementIndex>
5 |         <element attribute="Autor">
6 |             <topicAssoc>autor</topicAssoc>

```

Figura 7.10: Estrutura da definição de extração de associações *M2N* em XSTM

```

7      <role>escreve</role>
8    </element>
9    <element attribute="Artigo">
10      <topicAssoc>artigo</topicAssoc>
11      <role>escrito_por</role>
12    </element>
13  </members>
14 </M2N>

```

Essa especificação determina que o tópico do tipo *autor* que tiver o identificador idêntico ao caminho XPath `//Autoria/Autor` está associado com o tópico do tipo *artigo* que tem o identificador idêntico ao caminho XPath `//Autoria/Artigo`.

A Tabela 7.2 mostra como o processador XSTM-P interpreta o código acima. O conteúdo do identificador do tópico membro da associação deve ser igual ao conteúdo de sua referência na tabela de relacionamento.

Tabela 7.2: Associações *MxN*

	Identificador do tópico: //topic/xpath + //topic/xpath/@id	Referência na Tabela de Relacionamento: //members/elementIndex + //members/element/@attribute
Autor	//Inscricoes/Inscritos/Iniciais	//Autoria/Autor
Artigo	//Artigos/Artigo/@numero	//Autoria/Artigo

Assim, a partir da especificação *M2N* e do segmento de documento XML anteriormente apresentado, a seguinte associação seria gerada:

```

1  <association id="id_jcr_a15">
2    <instanceOf>
3      <topicRef xlink:href="#id_autor_artigo"/>
4    </instanceOf>
5    <member>
6      <roleSpec>
7        <topicRef xlink:href="#id_escreve"/>
8      </roleSpec>
9      <topicRef xlink:href="#id_jcr"/>
10   </member>
11   <member>
12     <roleSpec>
13       <topicRef xlink:href="#id_escrito_por"/>
14     </roleSpec>
15     <topicRef xlink:href="#id_a15"/>

```

```

16 |         </member>
17 |     </association>

```

Tal qual na definição de extracção dos tópicos, a definição de extracção das associações é um mapeamento de um para vários: pois apenas uma especificação define a extracção de várias associações. A quantidade de associações a ser geradas depende apenas do conteúdo do documento XML origem.

## 7.4 Trabalhos Relacionados com o TM-Builder

Actualmente, existem muitas ferramentas para apoio à criação de XTM, como, por exemplo, o Mapalizer (Algermissen, 2002). Contudo, não há nenhuma que através de uma especificação em XML (dos itens de informação relevantes) permite criar **Topic Maps** automaticamente usando apenas ferramentas XML.

Em uma das bibliografias mais recomendadas em termos de XML e metadados (Ahmed et al., 2001), há um capítulo sobre *Automated/Automatic Topic Map Construction*. Portanto, há mais de quatro anos o processo de construção automática de **Topic Maps** está em discussão. O capítulo em questão fornece 4 possíveis tipos de construção de **Topic Maps**:

1. Construção manual, a qual é a abordagem tradicional;
2. Combinação de construções manuais e automatizada;
3. O uso de uma ferramenta de construção visual baseada em XTM;
4. Reutilização de informação disponível em algum meio electrónico. Este é o caso seguido pelo **TM-Builder**, pois ele constrói o topic map a partir de recursos de informação baseado em XML.

O exemplo abordado em (Ahmed et al., 2001) demonstra os passos efectuados para a criação de um topic map em XTM a partir de uma fonte SVG (Ferraiolo and Jackson, 2003) e RDF Schema (Brickley and Guha, 2000b). Contudo, o exemplo apresentado está restrito a uma classe de documentos SVG e não propõe um extractor genérico, que tivesse a capacidade de extrair topic maps de qualquer tipo de documento. Além disso, o capítulo não especifica como a implementação do construtor de **Topic Maps** deve ser realizada.

Das ferramentas disponíveis para manipulação de **Topic Maps**, nenhuma é tão simples de instalar e usar como o **TM-Builder**. Praticamente todas as ferramentas relacionadas necessitam de outras tecnologias (interpretadores, programas utilitários ou bibliotecas) para executar suas tarefas. Como exemplos, citam-se:

- o *Ontopia Omnigator* (Ontopia, 2002b), o *TM4J* (SourceForge, 2003b) e o *Nexist* (SourceForge, 2003a) necessitam do *Tomcat* (Apache, 2003) instalado;

- o *SemanText*<sup>6</sup> que requer a instalação de um interpretador *Python* (Lutz and Ascher, 2003);
- o *TMTab – The Topic Map Tab* – necessita do *Protégé 2000* (Protégé, 2005) instalado.

O TM-Builder é uma ferramenta independente de outra tecnologia além de XML, pois depende apenas de um simples *parser* XML e de um processador de XSL para efectuar as transformações.

Outros trabalhos relacionados com o TM-Builder serão apresentados no Capítulo 8, pois tanto o TM-Builder quanto o Oveia são extractores de *Topic Maps*, e por isto, os trabalhos relacionados são os mesmos.

## 7.5 Sumário sobre TM-Builder

O objectivo deste capítulo foi a apresentação de uma arquitectura para a construção automática de extractores de *Topic Maps*, a partir do processamento de uma linguagem de especificação. Esses extractores, designados por TM-Builder, são o resultado de uma proposta abstracta, o *OntBuild* – discutido na Secção 7.1 – e são gerados a partir de uma especificação XSTM.

Em XSTM – linguagem para especificar a extracção de *Topic Maps* a partir de documentos XML – a definição da ontologia exige o mesmo esforço que em XTM; é necessária a especificação de cada um dos tipos de tópicos, dos tipos de associações e dos tipos de papeis de actuação em ocorrências.

Em XTM, tudo isso é visto como tópico. XSTM categoriza tais tópicos, dando-lhes uma semântica concreta maior, através da classificação como tipo de tópico, tipo de associação ou papel de actuação em associações. Então, do ponto de vista de descrição da ontologia, o ganho é obtido através de se passar a dispor de uma semântica mais precisa.

Para as instâncias extraídas dos recursos de informação, a situação é completamente diferente. Na especificação de tópicos e associações, usa-se expressões XPath que actuam como consultas. Desta forma, o ganho obtido é igual ao número de ocorrências recuperadas pela expressão de consulta: uma definição de tópico ou associação em XSTM recupera todas as instâncias do mesmo tipo.

O mais interessante desta proposta é que, por mais que aconteçam modificações no documento XML, incluindo-se novos elementos que não alterem a sua estrutura, não é necessário modificar a especificação XSTM. Sendo assim, o mesmo TM-Builder pode processar o documento XML alterado; ou seja, qualquer documento que pertença ao mesmo esquema XML.

---

<sup>6</sup><http://www.semantext.com/>





## Capítulo 8

# Oveia

*Gaste mais horas realizando que sonhando,  
fazendo que planejando,  
vivendo que esperando,  
porque embora quem quase morre esteja vivo,  
quem quase vive já morreu.*

Luiz Vernando Veríssimo

No Capítulo 6, apresentou-se um sistema integrado – chamado **Metamorphosis** – composto por vários módulos, que permite extrair informação de recursos, ou fontes, com heterogeneidade estrutural e semântica e criar uma rede de conhecimento navegável conceptualmente graças a uma ontologia de suporte.

O uso de ontologias para a explicitação do conhecimento implícito e oculto é uma possível abordagem para resolver o problema da heterogeneidade semântica. Em (Uschold and Gruninger, 1996), menciona-se interoperabilidade como uma aplicação chave de ontologias.

No Capítulo 7 descreveu-se a primeira solução para a extracção de informação e construção de um topic map com base numa dada ontologia. Concebida, implementada e testada essa primeira proposta, rápido se evoluiu para uma segunda geração de extractores, o **Oveia**, que ultrapassa a limitação do **TM-Builder**: só aplicável a documentos XML.

Assim, neste capítulo apresenta-se um extractor de dados e criador de topic maps dirigido por ontologias capaz de lidar com sistemas de informação heterogéneos, onde as fontes podem ser bases de dados relacionais ou documentos XML de famílias diferentes.

O capítulo inicia-se com a descrição da arquitectura do **Oveia** na Secção 8.1. A definição da linguagem **XSDS** (que serve para descrever as fontes) encontra-se na Secção 8.2 e se completa com seu processador, o *Extractor de Datasets* (Secção 8.3). A linguagem **XS4TM** (que define a ontologia e relaciona o topic map com as fontes) é descrita na Secção 8.4; o processador de **XS4TM** apresenta-se logo a seguir, na Secção 8.5.

O capítulo ainda mostra o processo de construção da **BD Ontologia** (Secção 8.6), apresen-

tando também quais as estratégias encontradas para o mapeamento de XML para bases de dados relacionais. A Secção 8.7 cita e compara os trabalhos relacionados com o Oveia e o TM-Builder. Por fim, o sumário do capítulo é realizado na Secção 8.8.

## 8.1 Arquitectura do Oveia

Como se disse acima, o Oveia representa uma evolução em relação ao TM-Builder, pois possui a capacidade de extrair topic maps a partir de recursos de informação que podem ser bases de dados relacionais ou documentos XML seguindo diferentes esquemas.

No TM-Builder, quando o recurso de informação não é composto por documentos XML, há a necessidade de uma conversão deste recurso para XML. Considerando que a maior parte dos recursos de informação, em empresas e instituições, estão armazenadas em bases de dados relacionais, torna-se óbvio que essas entidade estão perante um problema grave! Havendo a necessidade de gerar documentos XML com o conteúdo de tais base de dados, a informação passa a estar toda duplicada, não sendo possível garantir, ao longo do tempo, a coerência entre a base de dados e o documento XML; após cada alteração na base de dados, uma nova descarga da informação, para XML, necessita ser realizada.

Para evitar esta duplicação da informação e a preocupação com a coerência dos dados, desenvolveu-se um extractor de Topic Maps que acede directamente a ambos os tipos de recursos de informações. O Oveia permite que as informações sejam extraídas directamente do recurso de informação onde elas se encontram.

A arquitectura do Oveia pode ser expressada conforme os módulos visualizados na Figura 8.1, os quais estão abaixo descritos:

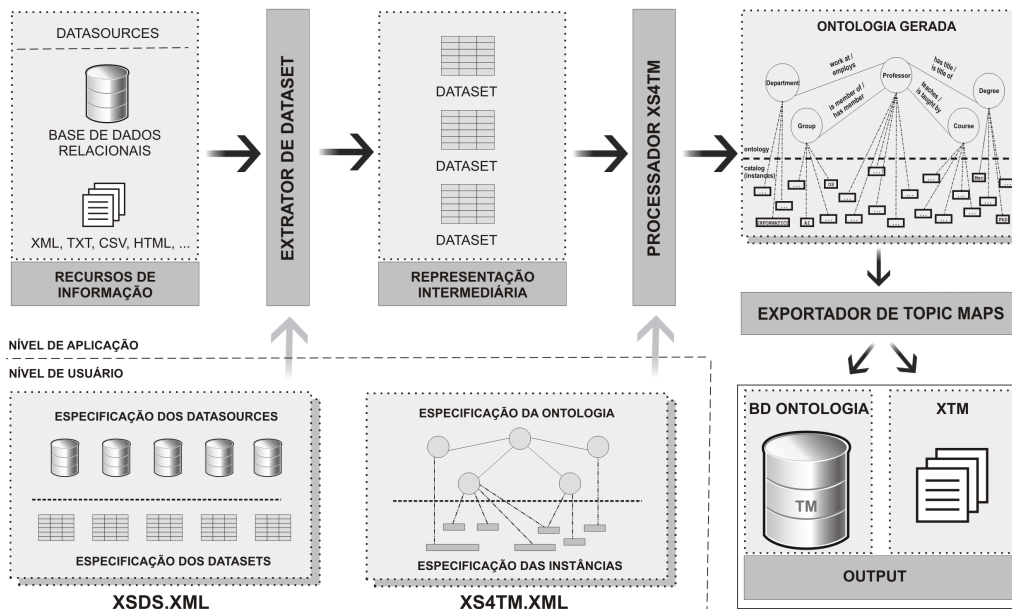


Figura 8.1: Arquitectura do Oveia

**Recursos de Informação:** são as fontes de dados (bases de dados ou documentos XML) que vão ser trabalhados pelo Oveia para extracção de dados. O seu conjunto constitui uma das entradas do extractor;

**Especificação XSDS:** é a definição formal, em sintaxe XML, de cada uma das fontes de informação a considerar;

**Extractor de Datasets:** é o processador responsável por extrair os dados dos recursos de informação especificados em XSDS e por armazená-los em **datasets**;

**Datasets:** é o conjunto dos dados extraídos dos recursos de informação heterogéneos, armazenados numa representação intermediária de formato homogéneo, prontos para serem mapeados para um topic map;

**Especificação XS4TM:** é a definição formal, num dialecto XML, do topic map que será construído a partir dos *datasets*, descrevendo a ontologia de suporte e a forma como esses conceitos abstractos se instanciam com os dados concretos e se ligam às fontes;

**Processador XS4TM:** é o processador responsável por interpretar a especificação XS4TM e, a partir dos dados contidos nos **datasets**, construir um topic map;

**Topic map gerado:** é um topic map representado em memória, gerado pelo processador XS4TM, o qual possibilita sua exportação para o formato escolhido pelo utilizador;

**Exportador de topic maps:** é o processador responsável por exportar para um formato definitivo o topic map gerado e guardado em memória. No Oveia, esse armazenamento pode ser realizado em dois formatos: num ficheiro de texto, com a sintaxe XTM; ou numa base de dados relacional (BD Ontologia).

As próximas secções descrevem as linguagens de especificação acima citadas e o seu processamento, além de descrever o modelo de dados da BD Ontologia.

## 8.2 XSDS: Especificação das Fontes de Dados

XSDS (*XML Specification for Data Sources*) é a linguagem definida com o intuito de especificar os recursos de informação que fornecerão dados para a criação dos **datasets**, os quais conterão a informação desejada para a formação dos **Topic Maps**. Essa especificação fornece todos os elementos necessários para determinar as fontes de dados passíveis de extracção de informação.

A especificação XSDS divide-se em duas partes:

1. definição dos recursos de informação (**Datasources**);
2. definição dos conjuntos de dados a serem extraídos (**Datasets**).

No início deste trabalho foi previsto a criação de uma linguagem universal para realizar consultas em recursos heterogéneos de informação. Mas a conversão das linguagens de consulta (SQL, XPath, ...) para a linguagem idealizada como universal não seria a solução ideal pelo facto de que o projectista teria a necessidade de aprender uma nova linguagem, ao invés de usar as linguagens amplamente conhecidas e difundidas. Dessa forma foi adoptada a estratégia de manter a linguagem de consulta de cada recurso.

Com o uso das linguagens de consulta específicas para cada tipo de recurso, define-se a criação de **datasets**. Assim, os **datasets** seriam o ponto de unificação de representação da informação entre os dados extraídos de cada fonte.

Nas próximas subsecções são apresentados os dois sub-conjuntos de XSDS: **Datasources** e **Datasets**. O primeiro define quais os recursos físicos de informação são definidos como entrada (*input*) para o Oveia; o segundo define quais conjuntos de dados serão extraídos destes recursos de informação que serão usados para a construção de **Topic Maps**, definindo assim a formação dos **datasets**.

### 8.2.1 Datasources: definição dos recursos de informação

Os **datasources** definem a localização física do recurso de informação. A declaração de cada um dos recursos de informação é realizada no elemento `<datasource>`, o qual define a conexão entre o extractor e os recursos de informação.

Os recursos são caracterizadas por um nome, que será usado como seu identificador, e por seus parâmetros de conexão com os recursos físicos de informação propriamente ditos. O elemento `<Datasources>` possui um atributo, chamado `@extractorDriver`, o qual indica o **driver** de extracção que será utilizado, de acordo com o tipo do recurso em questão.

A definição da arquitectura do extractor de **Topic Maps** foi idealizada para suportar extensão a diversos recursos de informação como fontes de dados. Para isso, essa arquitectura baseia-se no conceito de **drivers** de extracção.

O **driver** consiste em uma camada de abstracção capaz de tornar os dados transparentes para a especificação de extracção. Sendo assim, o **driver** fez um papel de tradutor entre as fontes de dados e o extractor Oveia.

No caso de uma base de dados, além da localização da mesma, são passados parâmetros como o utilizador e a palavra-passe para acedê-la, juntamente com o **driver** de extracção que fará este processo; no caso do recurso de informação ser um documento XML, é necessário apenas o nome do ficheiro com o seu caminho na árvore de directorias do sistema operativo.

### 8.2.2 Datasets: definição da extracção dos conjuntos de dados

Os **datasets** formam a estrutura intermediária de representação da informação referente às fontes de dados. A declaração de cada elemento `<dataset>` em XSDS contém a consulta a ser realizada para extrair as informações dos recursos.

Essa consulta é escrita na linguagem de consulta referente a cada recurso. Por exemplo, para bases de dados relacionais as consultas são escritas em SQL (*Structured Query Language*) (Cannan and Otten, 1992); para documentos XML, é utilizado o XPath (Clark and DeRose, 1999).

A estrutura do elemento `<dataset>` contém um identificador que será utilizado posteriormente em XS4TM para referenciar qual conjunto de dados está sendo processado.

O elemento `<dataset>` deve indicar qual `datasource` vai fornecer os metadados extraídos para a formação de cada `dataset`. Para cada elemento definido, um novo `dataset` será criado. Neste elemento, é necessário indicar a qual recurso esta a se referir.

O conteúdo do elemento `<dataset>` é uma expressão na linguagem de consulta referente ao tipo do recurso de informação. Caso este seja uma base de dados, o conteúdo deste elemento será uma expressão SQL para recuperar os metadados necessários para a construção do `dataset` em questão; se é um documento XML, o conteúdo deste elemento será uma expressão XPath, indicando o caminho para a informação deste `dataset`.

O `Dataset` garante a utilização do extractor para recursos distintos, pois esse mecanismo faz uma representação dos dados baseados em tabelas formadas por linhas e colunas, semelhante à representação relacional.

Finalizando, ressalta-se que o mesmo `datasource` pode ser fonte de dados na criação de vários `datasets`. Porém, um `dataset` somente será formado a partir de dados de um único `datasource`.

### 8.2.3 Sintaxe de XSDS

De um modo formal, a linguagem XSDS é definida pela *Gramática Independente de Contexto* (GIC) apresentada a seguir:

```

1 | Resources    ::= Datasources Datasets
2 | Datasources ::= @extractorDriver @name Datasource+
3 | Datasource  ::= Parameter+
4 | Parameter   ::= @name parameter
5 | Datasets    ::= Dataset+
6 | Dataset     ::= @name @database dataset? Columns? statement?
7 | Columns     ::= Column+
8 | Column      ::= @identify column

```

Assim como XSTM a linguagem XSDS é definida formalmente por um *Document Type Definition* (DTD) criado a partir da sua GIC. Assim, a estrutura dessa linguagem segue o DTD mostrado abaixo:

```

1 | <?xml version="1.0" encoding="UTF-8"?>
2 | <!ELEMENT resources (datasources, datasets)>
3 | <!ELEMENT datasources (datasource+)>
4 | <!ELEMENT datasource (parameter+)>
5 | <!--ATTLIST datasource
6 |      extractorDriver (br.uneb.dcet.tmbuilder.drivers.DataBase |
7 |                      br.uneb.dcet.tmbuilder.drivers.XMLFile) #REQUIRED

```

```

8      name CDATA #REQUIRED
9  >
10 <!ELEMENT parameter (#PCDATA)>
11 <!ATTLIST parameter
12     name (connectionURL | jdbcDriver | password | pathDocument | user) #REQUIRED
13 >
14 <!ELEMENT datasets (dataset+)>
15 <!ELEMENT dataset (#PCDATA | columns | statement)>
16 <!ATTLIST dataset
17     name CDATA #REQUIRED
18     database CDATA #REQUIRED
19 >
20 <!ELEMENT columns (column+)>
21 <!ELEMENT column (#PCDATA)>
22 <!ATTLIST column
23     identify CDATA #IMPLIED
24 >
25 <!ELEMENT statement (#PCDATA)>

```

Em um documento XML, de acordo com o DTD apresentado acima, especificam-se todos os recursos de informação que serão usados como fonte de informação em uma determinada extração de Topic Maps.

#### 8.2.4 Exemplo de Especificação em XSDS

A fim de facilitar o entendimento, segue um exemplo da especificação de um recurso de informação. No documento XSDS a seguir, são definidas três fontes de dados distintas:

1. uma base de dados Microsoft SQL Server® 2000, a qual é identificada pelo nome *BD\_DI\_MSSQL* (linhas: 7 a 14);
2. uma base de dados Oracle®, identificada pelo nome *BD\_DI\_ORACLE* (linhas: 16 a 21);
3. um documento XML, identificado pelo nome *BD\_DI\_XML* (linhas: 23 a 25).

Na definição de cada **datasource** é especificado o **driver** que será utilizado para interpretá-lo, além dos parâmetros que serão utilizados para a conexão com o recurso. Assim, é possível verificar que os **drivers** construídos para aceder os recursos estão sendo referenciados no caminho *//datasource/@extractorDriver*.

Em seguida, são definidos os **datasets**. Cada **dataset** referencia um **datasource**, o qual fora definido anteriormente. Através de SQL (no caso de base de dados) e XPath (no caso de documentos XML) define-se quais informações serão necessárias para a formação dos **datasets**. Portanto, esse exemplo demonstra a flexibilidade do Oveia, com a utilização de diferentes recursos.

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <!DOCTYPE resources SYSTEM "../..//dtds/datasources.dtd">
3 <resources>
4 <!-- ##### INÍCIO DAS DEFINIÇÕES DE DATASOURCES ##### -->

```

```

5 <datasources>
6 <!-- Declaração de uma Base de Dados Microsoft SQL Server -->
7 <datasource extratorDriver="br.uneb.dcet.tmbuilder.drivers.DataBase" name="BD_DI_MSSQL">
8   <parameter name="connectionURL">
9     jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=OVEIA
10   </parameter>
11   <parameter name="user">uxtm</parameter>
12   <parameter name="password">xtm</parameter>
13   <parameter name="jdbcDriver">com.microsoft.jdbc.sqlserver.SQLServerDriver</parameter>
14 </datasource>
15 <!-- Declaração de uma Base de Dados Oracle -->
16 <datasource extratorDriver="br.uneb.dcet.tmbuilder.drivers.DataBase" name="BD_DI_ORACLE">
17   <parameter name="connectionURL">jdbc:oracle:thin:@localhost:1521:orcl</parameter>
18   <parameter name="user">uxtm</parameter>
19   <parameter name="password">xtm</parameter>
20   <parameter name="jdbcDriver">oracle.jdbc.driver.OracleDriver</parameter>
21 </datasource>
22 <!-- Declaração de um Documento XML como Origem -->
23 <datasource extratorDriver="br.uneb.dcet.tmbuilder.drivers.XMLFile" name="BD_DI_XML">
24   <parameter name="pathDocument">C:\Uminho\DI.xml</parameter>
25 </datasource>
26 </datasources>
27 <!-- / ##### FIM DAS DEFINIÇÕES DE DATASOURCES ##### -->
28 <!-- ##### INÍCIO DAS DEFINIÇÕES DE DATASET ##### -->
29 <datasets>
30   <dataset name="DS_professor" database="BD_DI_MSSQL">
31     SELECT nome, email, site FROM professor
32   </dataset>
33   <dataset name="DS_aluno" database="BD_DI_MSSQL">
34     SELECT aluno_id,nome, email, rm, curso_id FROM aluno
35   </dataset>
36   <dataset name="DS_disciplina" database="BD_DI_MSSQL">
37     SELECT * FROM disciplina where status = 1
38   </dataset>
39   <dataset name="DS_calendario" database="BD_DI_ORACLE">
40     SELECT * FROM calendario_academico
41   </dataset>
42   <dataset name="DS_artigos" database="BD_DI_XML">
43     <columns>
44       <column identify="true">@ID</column>
45       <column>nome</column>
46       <column>idade</column>
47       <column>morada/rua</column>
48       <column>morada/cidade</column>
49     </columns>
50     <statement>//artigos/aluno</statement>
51   </dataset>
52 </datasets>
53 <!-- / ##### FIM DAS DEFINIÇÕES DE DATASET ##### -->
54 </resources>

```

A Figura 8.2 apresenta a relação entre os elementos `<datasources>` e `<datasets>`. O elemento `//datasources/@name` define o nome (identificador) do recurso de informação. Este nome será referenciado pelo elemento `//dataset/@database`, indicando de qual recurso de informação os dados serão extraídos para a formação do referido `dataset`.

De acordo com a Figura 8.2, o `dataset` com nome *DS\_aluno* vai ser construído a partir das informações contidas no `datasource` identificado pelo nome *BD\_DI\_MSSQL* e seu conteúdo serão os campos *aluno\_id*, *nome*, *email*, *rm*, *curso\_id* extraídos da tabela *aluno*.

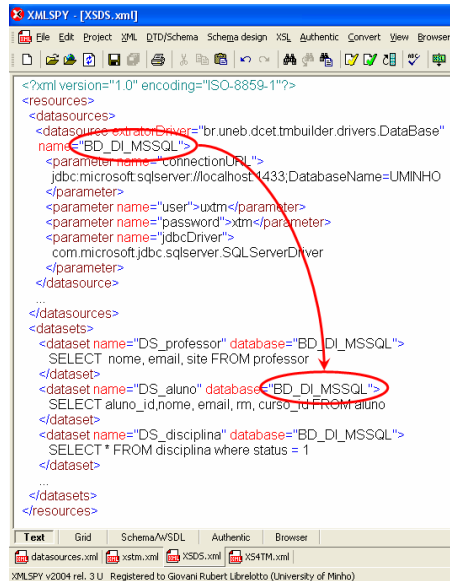


Figura 8.2: Relação entre **datasources** e **datasets** em XSDS

### 8.3 Extractor de Datasets

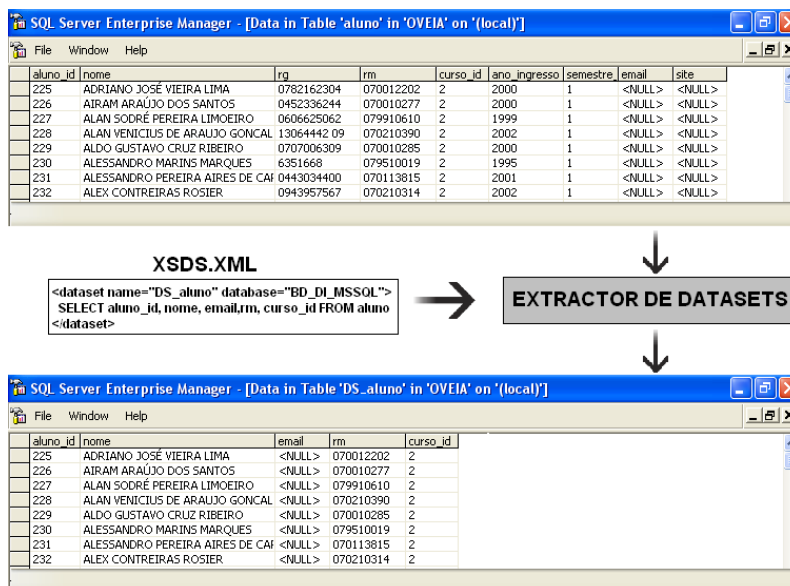
O *Extractor de Datasets* é um processador que extrai dados de recursos de informação (**datasources**) especificados em XSDS, gerando uma representação intermediária chamada **datasets**. Esta representação intermediária é composta por um conjunto de tabelas que contém a informação extraída das fontes de dados. Na prática, estas tabelas contém os dados seleccionados nos elementos **datasets** da especificação XSDS em questão.

A Figura 8.3 apresenta um exemplo da aplicação do *Extractor de Datasets*, baseado na especificação XSDS definida na secção anterior. A partir da tabela *aluno* da base de dados identificada por *BD\_DI\_MSSQL*, o processador selecciona os campos desejados – os quais serão necessários para a construção do topic map no processamento de XS4TM – criando um **dataset** chamado *DS\_aluno*.

A Figura 8.3 usa uma imagem capturada do módulo *Enterprise Manager* do *Microsoft SQL Server*® para representar as tabelas acima citadas. Apesar do **dataset** *DS\_aluno* estar representado como uma tabela do *Microsoft SQL Server*®, todo **dataset** criado pelo Oveia é armazenado em memória, em termos de objectos Java®, linguagem na qual este processador está implementado. Após o processamento de uma extracção de topic maps, os **datasets** são despejados da memória.

O *Extractor de Datasets* possui diversos **drivers** de extracção; um para cada tipo de recurso de informação. Actualmente, o Oveia possui dois **drivers** implementados: para base de dados relacionais e para documentos XML. A implementação de novos **drivers** para outros recursos de informação é um processo relativamente fácil e pode ser realizado conforme a necessidade. A construção de novos **drivers** define apenas um mapeamento da estrutura da fonte de informação (ou seja, do novo tipo de recurso a ser adicionado ao



Figura 8.3: Exemplo de funcionamento do *Extractor de Datasets*

Oveia) para os **datasets**, mas claro que a localização da fonte e a sua leitura terá que ser adaptada a cada caso.

Deste modo, esse modelo permite a criação de **drivers** de extracção para recursos específicos de informação; ou seja, além dos **drivers** implementados para bases de dados relacionais e documentos XML, outros **drivers** podem ser desenvolvidos, por exemplo, para documentos CSV, folhas de cálculos, páginas HTML, etc. O importante é que o projectista do novo **driver** tenha conhecimento da forma como a informação é armazenada em cada recurso, para assim, efectuar um mapeamento desta fonte para os **datasets**.

## 8.4 XS4TM: Uma linguagem XML para especificar a extracção de Topic Maps

A linguagem XSTM, proposta na Secção 7.3, foi inicialmente definida como sendo um dialecto XML para especificar o topic map que se pretende construir ao analisar documentos anotados pertencentes a um mesmo esquema XML. Por essa definição, o XSTM está directamente ligado a extracções a partir de documentos XML. Por outro lado, a necessidade de abranger novos recursos de informação fez com que se propusesse uma nova arquitectura para extracção de Topic Maps.

Dessa forma foi necessário repensar e redesenhar a linguagem XSTM, a qual passa a ser denominada por XS4TM. A especificação de extracção de ontologias em XS4TM tornou-se mais genérica e completa, fornecendo mecanismos para a especificação de todos os elementos do modelo de dados de Topic Maps (Garshol et al., 2003), o que não verificava-se em XSTM. Isso garante maior flexibilidade de especificação para diversos propósitos

de extracção, como pode-se comprovar através dos acréscimos da linguagem XS4TM em relação à XSTM abaixo listados:

- seguir a sintaxe XTM, tornando o processo de especificação intuitivo;
- permitir a especificação de indicadores de temas aos tópicos;
- permitir a inserção de tipos e contextos aos nomes;
- permitir o acréscimo de nomes variantes aos tópicos;
- permitir a definição de contextos para associações.

XSTM e XS4TM possuem o mesmo nome (*XML Specification for Topic Maps*) porque ambas foram projectadas como linguagens para especificação de extracção de Topic Maps.

#### 8.4.1 A sintaxe de XS4TM

A linguagem XS4TM tem por objectivo tornar a especificação da extracção de Topic Maps mais completa e flexível. XS4TM é caracterizado por transformar a actual norma XTM em um subconjunto da sua especificação.

Formalmente, XS4TM é representado pela *Gramática Independente de Contexto* abaixo:

```

1 XS4TM          ::= Ontologies Instances
2 Ontologies     ::= (Topic | Association)*
3 Instances      ::= (Topic | Association)*
4 Topic          ::= id (InstanceOf* SubjectIdentity? (BaseName | Occurrence)*)
5 InstanceOf     ::= id (TopicRef | ResourceRef | SubjectIndicatorRef)*
6 SubjectIdentity ::= id (TopicRef | ResourceRef | SubjectIndicatorRef)
7 TopicRef       ::= id xlink:type xlink:href
8 SubjectIndicatorRef ::= id xlink:type xlink:href
9 BaseName       ::= id (InstanceOf? Scope? BaseNameString Variant*)
10 BaseNameString ::= id baseNameString
11 Variant        ::= id (Parameters VariantName? Variant*)
12 VariantName    ::= id (ResourceRef | ResourceData)
13 Parameters     ::= id (TopicRef | ResourceRef | SubjectIndicatorRef)+
14 Occurrence     ::= id (InstanceOf? Scope? (ResourceRef | ResourceData))
15 ResourceRef    ::= id xlink:type xlink:href
16 ResourceData   ::= id resourceData
17 Association    ::= id (InstanceOf? Scope? Member+)
18 Member         ::= id (RoleSpec? (TopicRef | ResourceRef | SubjectIndicatorRef)*)
19 RoleSpec       ::= id (TopicRef | ResourceRef | SubjectIndicatorRef)
20 Scope          ::= id (TopicRef | ResourceRef | SubjectIndicatorRef)+

```

Cada especificação XS4TM é uma instância XML. Portanto, na prática a linguagem XS4TM é definida por um DTD (ou um XML-Schema), de modo a permitir o uso de todos os ambientes de processamento XML.

A especificação XS4TM é subdividida em duas partes distintas (ambas seguem o esquema de XTM 1.0 DTD):

**Ontologia:** A primeira parte é responsável pelas declarações de elementos responsáveis pela definição da ontologia, como os tipos de tópicos, os tipos de associações, ou qualquer outra definição de acordo com o modelo XTM que possa ser utilizada para expressar a estrutura do conhecimento a ser extraído;

**Instâncias:** A segunda parte representa a extracção das instâncias de tópicos e associações a partir das fontes de informação. Nesse momento, os **datasets** serão utilizados para expressar quais recursos de informação fornecerão os dados para a construção de tópicos e associações.

A linguagem XS4TM possui assim dois elementos principais: `<ontologies>` e `<instances>`. Cada elemento possui sua estrutura definida de acordo com a sintaxe XTM. Isto significa que esses elementos são compostos por elementos `<topic>` e `<association>`.

A única diferença encontra-se nos elementos `//instances/topic` e `//instances/association`, pois estes possuem um atributo denominado `@dataset`, utilizado para definir a partir de qual **dataset** o tópico ou a associação será construído. Este atributo é uma referência ao nome do **dataset**, declarado no documento de especificação XSDS.

A Figura 8.4 apresenta um exemplo desta referência. O **dataset** chamado *DS\_aluno* é a fonte de dados para a formação dos tópicos do tipo *aluno*.

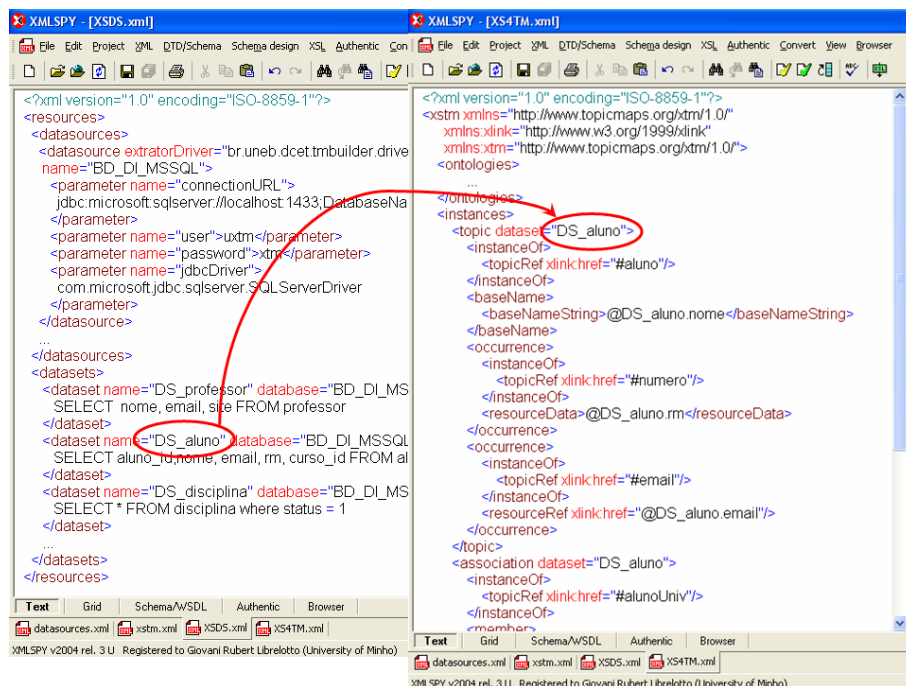


Figura 8.4: Relação entre XSDS e XS4TM

De forma a ilustrar a utilização e a aplicabilidade da nova especificação, a próxima subsecção apresenta, de forma resumida, documento XS4TM com a declaração do tópico *aluno* no conteúdo do elemento *ontologies*.

### 8.4.2 Exemplo de Especificação em XS4TM

Seguindo o mesmo exemplo, encontra-se a ilustração do uso do dataset *DS\_aluno*, o qual permite declarar instâncias do tópico em questão; assim, todos os tópicos criados a partir deste dataset serão instâncias do tópico *aluno* (conforme a linha 17). A seguir, os dados contidos no dataset *DS\_aluno* podem ser utilizados para o preenchimento das características dos tópicos criados.

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <xs4tm xmlns="http://www.topicmaps.org/xtm/1.0/"
3   xmlns:xlink="http://www.w3.org/1999/xlink"
4   xmlns:xtm="http://www.topicmaps.org/xtm/1.0/">
5   <ontologies>
6     <topic id="aluno">
7       <baseName>
8         <baseNameString>Aluno</baseNameString>
9       </baseName>
10    </topic>
11    ...
12  </ontologies>
13  <instances>
14    <topic dataset="DS_aluno">
15      <instanceOf>
16        <topicRef xlink:href="#aluno"/>
17      </instanceOf>
18      <baseName>
19        <baseNameString>@DS_aluno.nome</baseNameString>
20      </baseName>
21      <occurrence>
22        <instanceOf>
23          <topicRef xlink:href="#numero"/>
24        </instanceOf>
25        <resourceData>@DS_aluno.rm</resourceData>
26      </occurrence>
27      <occurrence>
28        <instanceOf>
29          <topicRef xlink:href="#email"/>
30        </instanceOf>
31        <resourceRef xlink:href="@DS_aluno.email"/>
32      </occurrence>
33    </topic>
34    ...
35  </instances>
36 </xs4tm>

```

Para o preenchimento das informações referentes a cada tópico, é necessário buscar tal informação no dataset que a contém. Assim, identifica-se essas propriedades com a expressão:

@ + "dataset" + "." + "atributo"

Em resumo, isto significa que:

- O @ apenas indica que esta declaração é referente a uma propriedade de um dataset;

- Entre o @ e o ponto, encontra-se o identificador do **dataset** (especificado em XSDS) ao qual deseja-se recuperar a informação. No exemplo da Figura 8.5, o **dataset** seleccionado é o *DS\_aluno*.
- O **atributo** é uma referência ao campo do **dataset** que contém a informação desejada. Na Figura 8.5, o atributo recuperado para a construção da ocorrência do tipo *email* é o campo *email* extraído pelo **dataset** *DS\_aluno*.

Então, a referida expressão é assim representada:

@DS\_aluno.email

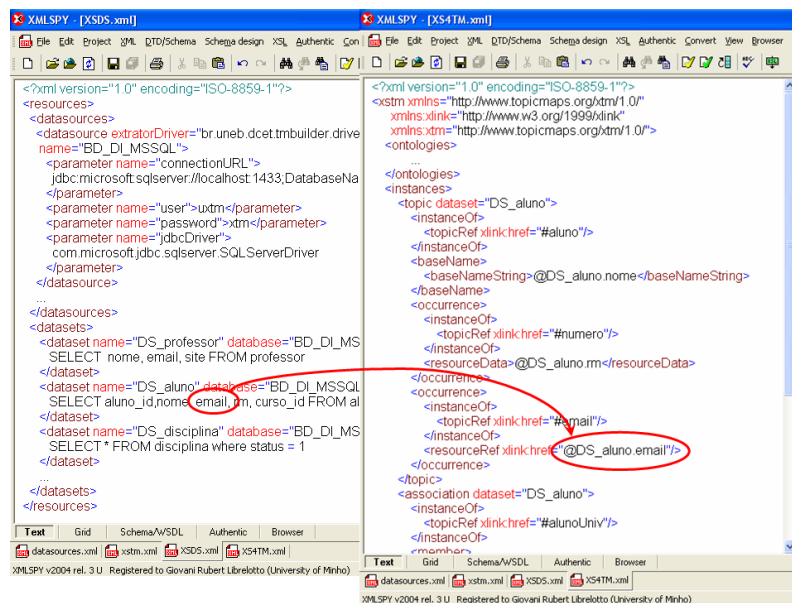


Figura 8.5: Relação entre XSDS e XS4TM

Desta forma, habilita-se o uso das informações contidas nos **datasets**, de uma forma directa.

## 8.5 Processador XS4TM

Este componente utiliza a especificação XS4TM para seleccionar quais campos dos *datasets*, extraídos dos recursos de informação, são necessários para a formação do topic map. Este processador é um interpretador que tira vantagem da organização das informações em um formato uniforme.

O seu processo de execução pode ser resumido em três passos:

1. ler a especificação XS4TM e extrair os dados especificados que encontram-se nos **datasets**;

2. criar o topic map baseado na própria especificação XS4TM;
3. armazenar o topic map gerado na BD Ontologia ou em um documento no formato XTM.

O armazenamento de um topic map no formato XTM é algo trivial, pois a estrutura deste tipo de documento é bem definida. Entretanto, para o armazenamento em bases de dados relacionais, primeiramente tem-se que definir um modelo correspondente aos **Topic Maps**. A próxima secção apresentará o modelo desenvolvido para o armazenamento dos **Topic Maps** gerados pelo Oveia.

## 8.6 Armazenamento de Topic Maps em base de dados relacionais

No momento em que se definiu que os **Topic Maps** gerados pelo Oveia seriam também armazenados em bases de dados relacionais, partiu-se para a especificação de um modelo que congregasse todas as características encontradas num topic map no formato XTM.

Para determinar como armazenar um topic map em base de dados relacionais, verificou-se inicialmente as estratégias existentes para armazenamento de documentos XML em bases de dados, porque o problema encontrado justamente era esse: qual seria a estrutura de uma base de dados para conter o conteúdo de um topic map, a qual já encontrava-se definida numa sintaxe XML?

Deste modo, esta secção apresenta duas estratégias de mapeamento de XML para base de dados. Logo após, encontram-se os argumentos da decisão tomada, descrevendo qual estratégia foi adoptada para armazenar os **Topic Maps** em base de dados e o porque da decisão.

### 8.6.1 Armazenamento de XML em Base de Dados

Armazenar um documento XML em uma base de dados XML nativa é conceptualmente mais simples do que converter o modelo XML para qualquer um dos outros modelos de bases de dados existentes. No entanto, devido à imaturidade e ao elevado custo dos sistemas nativos actuais, a maioria dos sistemas que armazenam XML utilizam uma base de dados relacional ou orientada a objectos.

Sendo assim, a opção por modelação da norma **Topic Maps** para o modelo relacional foi adoptada, também por levar em consideração o facto de que a maioria dos utilizadores (sejam particulares, sejam empresariais) trabalham com bases de dados relacionais.

O mapeamento de XML para o modelo relacional pode-se dividir em duas categorias (Yoshikawa, Amagasa, and Uemura, 2001):

**Mapeamento por estrutura:** o esquema das tabelas relacionais é derivado do DTD;

**Mapeamento por modelo:** o esquema das tabelas relacionais é genérico e, portanto, independente de qualquer DTD.

### Mapeamento por Estrutura

Nesta estratégia, o desenho da base de dados relacional é baseado na estrutura do documento XML que se pretende armazenar, tipicamente definida por um DTD ou XML-Schema.

Dado um DTD arbitrário, é possível seguir 18 regras algorítmicas para chegar a um modelo de tabelas relacionais (Williams et al., 2000). Estas tabelas suportam o armazenamento de qualquer documento XML que respeite a estrutura definida no esquema original. Contudo, este procedimento é demorado e repetitivo; por isso existem produtos que tentam automatizar este processo, tais como Microsoft SQLXML (Microsoft, 2003) e o Oracle XSU (Oracle, 2002).

É importante referir que nem todos os produtos suportam todas as características desejáveis, não sendo capazes de processar esquemas com complexidade elevada. Além disso, essas ferramentas deveriam ser capazes de transformar pesquisas em uma linguagem tipo XML (por exemplo, XPath) em instruções de SQL para o sistema RDBMS (*Relational Database Management System*) subjacente; porém, nem todos suportam essa funcionalidade.

Dado que o repositório a implementar tem que suportar um número arbitrário de DTD's diferentes, e se para cada DTD fossem criadas cerca de 50 a 100 tabelas de suporte para essa estrutura específica, o número de tabelas a gerir seria provavelmente na ordem das várias dezenas de milhares.

### Mapeamento por Modelo

A segunda estratégia para o mapeamento de XML para o modelo relacional consiste no chamado mapeamento por modelo (Yoshikawa, Amagasa, and Uemura, 2001) em que as tabelas são criadas, não para suportar um DTD em particular, mas sim para suportar qualquer documento XML. Por outras palavras, a partir do modelo XML (constituído por elementos, atributos, CDATA, etc.) são criadas tabelas genéricas para armazenar todos os tipos de documentos, qualquer que seja o DTD.

Em complemento a estas duas estratégias que aproveitam o que as bases de dados relacionais oferecem, surgiu uma nova forma de armazenamento: o mapeamento em coluna, onde o sistema de gestão de base de dados relacional suporta um novo tipo de dados para armazenar XML, o que se explica a seguir.

### Armazenamento em Coluna

Para além dos modelos básicos de mapeamento para relacional, alguns vendedores de bases de dados relacionais optaram por estender os seus produtos, oferecendo um novo tipo de

dado capaz de armazenar numa coluna, ou campo, todo documento XML; sendo possível efectuar pesquisas e até suportar a definição de índices. Como exemplo, pode-se citar o Oracle 9i<sup>1</sup> que suporta este tipo de armazenamento através do tipo de dados `XMLType`<sup>2</sup>.

### 8.6.2 Solução Adoptada: BD Ontologia

A estratégia escolhida – no contexto do Oveia, para armazenar o topic map extraído – foi baseada no conceito de mapeamento por estrutura. Para tomar esta decisão foram analisadas as vantagens e desvantagens dos mapeamentos apresentados.

A opção de utilizar as ferramentas de mapeamento automatizado, implicaria que existisse uma grande confiança na capacidade do produto automatizar completamente todo este processo de mapeamento de estruturas XML arbitrariamente complexas para tabelas relacionais. Desta forma foi descartada esta opção, adoptando-se o processo manual de transformação do modelo XML para o modelo relacional.

No caso de adoptar um mapeamento por modelo implicaria em consultas mais complexas para recuperar as informações extraídas, tornando esta tarefa muito trabalhosa. Além disso, um modelo relacional normalizado é mais fácil de ser compreendido.

A hipótese de mapeamento por estrutura se mostrou mais adequada, pois todos os documentos a armazenar correspondem a um único DTD, bastante bem conhecido. Desta forma, a tarefa foi transformar os elementos definidos em XTM 1.0 DTD (Pepper and Moore, 2001b) em entidades na base de dados relacional, preservando suas características e relacionamentos.

A representação de Topic Mapsem base de dados relacional é chamada neste trabalho de BD Ontologia. Esta é a responsável por manter os Topic Maps extraídos.

### Modelação e características da BD Ontologia

Um dos diferenciais desta ferramenta é o armazenamento dos Topic Maps extraídos em uma base de dados relacional. De acordo com a discussão da Secção 8.6.1, referente aos métodos de mapeamento de documentos XML para o modelo relacional, adoptou-se na BD Ontologia o modelo de mapeamento por estrutura.

De acordo com essa estratégia, foi criada uma tabela para cada elemento do XTM 1.0 DTD (Pepper and Moore, 2001b). Esse processo pode ser entendido facilmente analisando o fragmento do XTM 1.0 DTD abaixo e a Figura 8.6, que mostra as tabelas correspondentes.

```

1 | <!ELEMENT topic (instanceOf*, subjectIdentity?, (baseName | occurrence)*)>
2 | <!--ATTLIST topic
3 |   id ID #REQUIRED
4 | >
5 | <!--ELEMENT subjectIdentity (resourceRef?, (topicRef | subjectIndicatorRef)*)>

```

<sup>1</sup><http://www.oracle.com/ip/deploy/database/oracle9i/>

<sup>2</sup><http://www.oracle-base.com/Articles/9i/XMLTypeDatatype.php>



```

6 <!--ATTLIST subjectIdentity
7   id ID #IMPLIED
8 >
9 <!--ELEMENT baseName (scope?, baseNameString, variant*)>
10 <!--ATTLIST baseName
11   id ID #IMPLIED
12 >
13 <!--ELEMENT occurrence (instanceOf?, scope?, (resourceRef | resourceData))>
14 <!--ATTLIST occurrence
15   id ID #IMPLIED
16 >

```

O trecho do DTD de XTM apresentado acima define a estrutura dos tópicos (elemento **topic**) em um topic map. Como visto acima, um tópico possui uma ou mais instâncias, zero ou uma identidade de tema, assim como zero ou mais nomes e ocorrências.

No modelo relacional, o elemento **topic** foi mapeado para uma tabela, a qual contém seu identificador (**topicID**), uma referência ao seu identificador de tema (**subjectIdentityID**) e o identificador a qual topic map este tópico pertence (**topicMapID**). Esse último campo é necessário por uma mesma BD Ontologia pode armazenar vários topic maps ao mesmo tempo.

Também foi criada uma tabela para definir as instâncias de cada tópico, chamada **instanceOfTopic**. Esta tabela contém os identificadores de cada relação hierárquica: o do pai (**instanceOfID**) e o do filho (**topicID**).

A tabela **baseName**, além do identificador (**baseNameID**) e do nome propriamente dito (**baseNameString**), possui campos para identificar a qual tópico ele pertence (**topicID**) e em que contexto ele está inserido (**scopeID**).

Por sua vez, a tabela **subjectIdentity** armazena todos os identificadores de temas, contendo seu identificador (**subjectIdentityID**) e o recurso que está sendo referenciado (**resourceRefID**).

Finalmente, a tabela **occurrence** armazena os recursos externos apontados pelos tópicos. O campo **occurrenceID** é o identificador de cada ocorrência, enquanto que **scopeID** é uma referência o seu contexto, **topicID** é uma referência ao tópico a que esta ocorrência pertence e **instanceOfID** indica qual é o seu tipo. Os campos **resourceRefID** e **resourceData** contém a ocorrência propriamente dita, ou seja, a informação a qual é uma característica do tópico indicado em **topicID**.

Os relacionamentos citados acima, baseados no DTD de XTM, são apresentados graficamente na Figura 8.6.

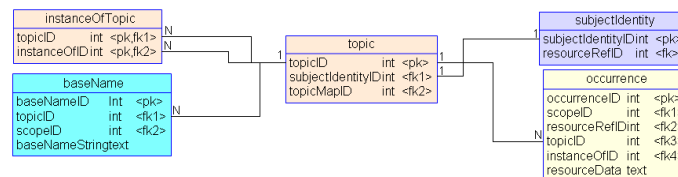


Figura 8.6: Trecho do Modelo ER do BD Ontologia

A facilidade de compreensão desse modelo é garantida principalmente pelo facto de que este modelo segue a norma XTM, a qual é bastante conhecida pela comunidade académica. Essa foi umas das vantagens trazidas por essa opção de modelação, preservando a norma Topic Maps.

Como ilustrado na Figura 8.1 (arquitetura do Oveia), os dados extraídos dos recursos de informação são representados como **datasets** e transformados pelo processador para, em seguida, serem armazenados na BD Ontologia.

## 8.7 Trabalhos Relacionados com o Oveia

Nesta secção pretende-se realizar uma apresentação das principais ferramentas para extracção de ontologias com o objectivo de estabelecer uma comparação do Oveia com o KAON REVERSE (Volz, 2003), o *OntoExtract* (Vieira, Tanaka, and Moura, 2002) e o *SemanTag* (Fröhlich and Schmuck, 2004).

O KAON (Handschuh et al., 2001) é um projecto **open-source** que fornece uma infraestrutura para gestão de ontologias voltado para aplicações de negócios. A ferramenta KAON REVERSE é um *plug-in* do *framework* KAON<sup>3</sup>. O KAON REVERSE permite a conexão e o mapeamento de bases de dados relacionais para uma ontologia, tendo o objectivo de extrair instâncias e relacionamentos de instâncias a partir da base de dados. Dentre as ferramentas conhecidas, esta é a que mais se aproxima do Oveia.

A Tabela 8.1 compara as características e funcionalidades do Oveia com o KAON REVERSE. Os dados presentes nesta tabela foram baseados na análise feita em (Vieira, 2002).

Tabela 8.1: Comparativo entre KAON REVERSE e Oveia

	KAON REVERSE	OVEIA
Linguagem	Java	Java
Uso de APIs	Sim	Sim
Uso de Engenharia Reversa	Sim	Não
Especificação	Árvore(GUI)	Documento XML
Fontes de Extracção de Ontologias	BDs relacionais via JDBC	BDs relacionais via JDBC, XML, extensível a outras fontes
Norma de Representação de Ontologias	RDF	Topic Maps
GUI (Interface Gráfica)	Sim	Não
Resultado Gerado	Documento RDF	Base de Dados de Ontologias

De acordo com a Tabela 8.1, é perceptível as vantagens de cada ferramenta em pontos distintos. Seria mais sensato dizer que existe uma certa tendência à união das funcionalidades, do que comparar qual seria a melhor ferramenta. Partindo deste ponto de vista, destaca-se as vantagens de cada uma. Por um lado, a KAON REVERSE apresenta van-

<sup>3</sup>Mais informações em: <http://kaon.semanticweb.org/>

tagens em relação ao uso de interface gráfica para a especificação de ontologias e no uso de engenharia reversa para auxiliar o mapeamento dos recursos de informação. Por outro lado, o **Oveia** se destaca por ser mais flexível em relação aos recursos passíveis de extracção e em relação ao processo de especificação, porque segue uma sintaxe normalizada – XTM. Além disso, o **Oveia** se diferencia por gerar uma base de dados ontológica capaz de manter o topic map extraído.

Em relação à norma de representação, as duas ferramentas possuem focos diferentes. Enquanto uma faz uso de RDF, a outra utiliza **Topic Maps**. Na Secção 3.5.5 foram apresentados argumentos que levaram a optar pelo uso da norma **Topic Maps**. A opção pela norma **Topic Maps** assentou, principalmente, no seu poder de representatividade – superior ao do RDF – e na possibilidade de criação de navegadores conceptuais.

Ao fazer uso de uma linguagem de especificação do processo de extracção semelhante à norma **Topic Maps** (XS4TM), o **Oveia** facilita a tarefa do projectista da ontologia, sendo esta uma vantagem considerável. Para cada caso, poderá ser criada uma visão diferente da fonte de dados, bastando para isso expressar uma nova estrutura de conhecimento específica.

Em relação a outros extractores de ontologias, pode-se citar o *OntoExtract*. O *OntoExtract* extrai ontologias a partir do esquema de uma base de dados relacional, como citado em (Vieira, Tanaka, and Moura, 2002). Este extractor apresenta a proposta de definir o RDF como linguagem base para definição de objectos de ontologias, como uma espécie de camada de modelo; enquanto que o XTM seria utilizada como uma linguagem de definição de ontologias, uma (meta)ontologia.

Assim como o KAON REVERSE, o *OntoExtract* também só suporta a extracção em bases de dados, fazendo a engenharia reversa do esquema relacional. Limitam, assim, a faixa de recursos de informação passíveis de extracção.

O **SemanTag** (Fröhlich and Schmuck, 2004) – *Topic Map Generator* – é uma ferramenta para geração automática de topic maps a partir de diferentes fontes de dados utilizando definições de regras especificadas na sintaxe *Jelly XML* (Apache, 2005). É uma biblioteca Java® que fornece elementos *Jelly* que permitem aceder e modificar fontes de dados para criar uma rede semântica. Contudo, **SemanTag** é ainda um projecto em desenvolvimento, não existindo muito material publicado sobre ele.

## 8.8 Sumário sobre o Oveia

O objectivo deste capítulo foi a apresentação de uma arquitectura para a construção automática de **Topic Maps**, a partir da extracção de dados de recursos heterogéneos de informação. Esse sistema, designado por **Oveia**, resultou de uma proposta inicial denominada TM-Builder, apresentada em (Librelotto, Ramalho, and Henriques, 2003d) e descrita no Capítulo 7, que não é mais do que um extractor de ontologias para documentos XML.

A extracção de informação para a construção de **Topic Maps** a partir de recursos heterogéneos de informação é especificada pela linguagem XS4TM, a qual define os conceitos

e relações que se podem encontrar nestes recursos, os quais serão mapeadas para tópicos e associações, respectivamente.

A nova arquitectura, a fim de ultrapassar a limitação do tipo de fonte suportada, baseia-se no conceito de **drivers** para a extracção da informação. Um **driver** é um processador específico para a extracção de dados de um recurso de informação específico.

As principais vantagens desta proposta são as possibilidades de adaptação para o processo de extracção de ontologias. Em um primeiro plano, o Oveia permite a adição de novos recursos de informação, de acordo com as necessidades encontradas pelo projectista. Assim, para o desenvolvimento de novos *drivers*, basta um mapeamento do recurso para **datasets**.

Outra vantagem é o facto de que a ontologia especificada em XS4TM não depende da quantidade de informação extraída dos recursos de informação. Além disso, é importante frisar que modificações nos recursos de informação (obviamente mudanças ao nível de seu conteúdo, e não mudanças estruturais) não acarretam mudanças na especificação XS4TM; basta voltar a invocar o Oveia que o novo topic map (reflectindo as alterações) é de imediato reescrito.

## Capítulo 9

# XTche

*É o meu Rio Grande do Sul, céu, sol, sul, terra e cor  
onde tudo que se planta cresce  
e o que mais floresce é o amor*

Leonardo

Como apresentado no Capítulo 5, no desenvolvimento de topic maps reais é altamente conveniente utilizar um sistema para validá-los; isto é, verificar a correcção de uma instância real de acordo com a especificação formal da respectiva família de topic maps (de acordo com a intenção de seu criador).

Os **Topic Maps** podem ser representados em diversas linguagens, mas o processo de validação deve poder manipular qualquer uma dessas representações sintácticas. Porém um validador automático de restrições exige uma representação específica de **Topic Maps** processável por máquina. Portanto, para satisfazer o requisito inicial é necessário que os vários formatos distintos possam ser mapeados para a representação suportada pelo validador.

Para a decisão de que formato de representação de **Topic Maps** utilizar, levou-se em conta o que a comunidade académica tem adoptado. A sintaxe XTM é, sem dúvida, a linguagem mais utilizada, além de ser a única que é uma norma internacional (HyTM também é uma norma, contudo é considerada ultrapassada). Além disso, todas as outras notações usadas para descrever **Topic Maps** possuem processadores que as transformam em XTM e vice-versa (Ogievetsky, 2000; Barta, 2002). Isso justifica a escolha de XTM como linguagem para escrita dos topic maps que vão ser validados.

As restrições em topic maps também podem ser representadas em diversas formas, portanto um formato comum também é fundamental.

A representação das regras semânticas não necessariamente tem que seguir a sintaxe XTM; a principal justificativa para não se ter adoptado XTM neste projecto é que tal linguagem é muito verbosa. Portanto, uma sintaxe que permita a definição de restrições de uma forma mais compacta e eficiente torna-se fundamental.

Desta forma, o presente capítulo apresenta uma linguagem – e seu respectivo processador – para a definição de restrições sobre **Topic Maps**, chamada **XTche**, que permite garantir que um conjunto de topic maps de uma mesma família<sup>1</sup> são semanticamente válidos de acordo com uma especificação.

**XTche** faz parte de um conjunto de linguagens para processamento de **Topic Maps**, pertencentes ao **Metamorphosis**. Este ambiente, que foi introduzido no Capítulo 6, permite a definição, extracção, validação e navegação de topic maps a partir de recursos heterogêneos de informação.

No que diz respeito a validação de **Topic Maps**, uma lista de requisitos para *Topic Map Constraint Language* (TMCL) (Nishikawa, Moore, and Bogachev, 2004) foi recentemente estabelecida pelo *ISO Working Group* – o projecto ISO JTC1 SC34 – a qual é exaustiva e cobre praticamente todos os objectos de **Topic Maps**, conforme foi apresentado na Secção 5.4. A linguagem **XTche**, aqui proposta e considerada o contributo fundamental deste trabalho de doutoramento, atende praticamente todos os requisitos dessa lista; para este propósito, **XTche** possui um conjunto de construtores para a descrição de restrições em **Topic Maps**, o qual será detalhado nas próximas secções.

Uma especificação **XTche** assenta num esquema básico que é genérico, porém incompleto; assim, deve ser preenchido com as restrições que serão apresentadas na Secção 9.1, as quais podem ser de três tipos: **restrições contextuais**, **restrições de esquema** e **restrições de existência**.

A Secção 9.2 apresenta a linguagem **XTche**, a qual é baseada em XML Schema. A referida secção fornece, então, o esquema básico desta linguagem, assim como elementos e atributos necessários para uma completa especificação de restrições em **Topic Maps**.

Os tipos de restrição descritos na Secção 9.1 são detalhados na Secção 9.3, onde os tipos de restrições previstos por TMCL são classificados nos tipos de restrições de **XTche**.

Uma das principais inovações desta proposta, detalhada na Secção 9.4, é incluir a definição do esquema da família de topic maps em causa na própria especificação **XTche**. Assim torna-se possível definir a estrutura que os tópicos, associações e ocorrências – de um determinado topic map – devem obedecer, na mesma especificação em que se descreve as restrições (usando, portanto, a mesma sintaxe).

O Processador **XTche** toma como entrada a especificação **XTche**, gerando um validador de **Topic Maps** (TM-Validator). Os detalhes sobre este processo encontram-se na Secção 9.5.

O processo de validação de uma especificação **XTche** é mostrado na Secção 9.5.2: pois antes de seu processamento (transformação num validador semântico), a sua correcção deve ser verificada. Essa correcção é testada em duas etapas: primeiro confere-se se a especificação **XTche** é um XML Schema, para posteriormente, garantir que este XML Schema pertence à linguagem **XTche**.

---

<sup>1</sup>Por topic maps de uma mesma família entende-se topic maps que descrevam um mesmo tipo de domínio. Por exemplo: um topic map sobre a *Universidade do Minho* e outro topic map sobre a *Universidade Federal do Rio Grande do Sul* são da mesma família, pois ambos descrevem o domínio “universidade” e possuem características comuns, como os mesmos tipos de tópicos, tipos de associações, contextos, etc.

As linguagens para especificação de restrições em Topic Maps conhecidas até o momento são comparadas na Secção 9.6, onde se apresentam AsTMa! e OSL como os principais trabalhos relacionados com XTche.

Por fim, o sumário que se faz na Secção 9.7 finaliza o capítulo, resumindo as ideias aqui apresentadas.

## 9.1 Classificando as restrições sobre Topic Maps

As restrições da linguagem XTche são divididas em três partes: **restrições de esquema** (*schema constraints*), **restrições contextuais** (*contextual constraints*) e **restrições de existência** (*existencial constraints*).

O primeiro sub-conjunto é composto pelas regras que permitem a definição da estrutura dos tópicos, associações e ocorrências, isto é, o *Topic Map Schema*. Essas regras determinam quais as características que podem ou devem estar presentes em tópicos e associações. Definido a partir de *Topic Map Data Model* (Garshol et al., 2003), as **regras de esquema** cobrem todas as possíveis verificações que devem ser realizadas sobre um topic map. As restrições deste tipo estão abaixo listadas. A Subsecção 9.3.1 descreve o modo de especificação deste tipo de restrição.

### Restrições de Esquema :

- Tópico do tipo T deve ter um número específico de nomes/ocorrências/indicadores de tema (cardinalidade);
- Tópico do tipo T deve ter como nomes/ocorrências/indicadores de tema um valor de acordo com um padrão em particular;
- Tópico do tipo T (não) deve ter nome/ocorrência/indicador de tema;
- Tópico do tipo T (não) deve ter um nome/ocorrência no contexto S;
- Tópico do tipo T (não) deve ter um nome/ocorrência, o qual é instância de tipo de tópico T, no contexto S;
- Tópico do tipo T (não) deve ter um nome que é do tipo N;
- Tópico do tipo T (não) deve ter uma ocorrência que é do tipo O;
- Tópico T (somente/não) pode ser usado como um papel de actuação em uma associação do tipo A;
- Tópico do tipo T (somente/não) pode ser usado como um actor em uma associação do tipo A;
- Uma lista dos tópicos que devem ser instâncias do tipo de tópico T;
- Tópico do tipo T deve ter X instâncias;
- Toda associação instância de A deve estar inserida no contexto S;
- Toda associação instância de A (somente/ao menos/não) possui os papéis de actuação R1 e R2;

- Toda associação instância de A tem (somente/ao menos/não) dois tópicos actores, onde o primeiro é do tipo T1 e o segundo é do tipo T2;
- Toda associação instância de A (não) deve ter o papel R sendo actuado por um tópico do tipo T;
- Toda associação instância de A tem o papel R sendo actuado por X tópicos do tipo T (cardinalidade);
- Associação instância de A tem o papel R1 sendo actuado por um tópico do tipo T1 e o papel R2 sendo actuado por tópico do tipo T2 ou T3;
- Toda associação instância de A deve ter dependência entre os tipos dos tópicos actores;
- Ocorrência do tipo O (somente/não) pode ser característica de tópicos do tipo T;
- Ocorrência do tipo O (somente/não) pode ser usado no contexto S;
- Ocorrência do tipo O deve ter apontadores que combinem com um URI, de acordo com um padrão P.

O segundo sub-conjunto, por sua vez, é aplicado sobre restrições particulares num topic map. A partir deste tipo de restrições, restringe-se as situações que um tópico pode ser encontrado num topic map, tal como será demonstrado na lista de regras abaixo que fazem parte deste tipo. O modo de especificação das regras deste tipo será apresentado na Subsecção 9.3.2.

#### **Restrições Contextuais :**

- Tópico T (somente/não) pode ser usado como tipo de tópico;
- Tópico T (somente/não) pode ser usado para ser tipo de nomes;
- Tópico T (somente/não) pode ser usado para ser tipo de ocorrências;
- Tópico T (somente/não) pode ser usado para ser tipo de associações;
- Tópico T (somente/não) pode ser usado para definir contexto de associações;
- Tópico T (somente/não) pode ser usado para definir papel de actuação em associação;
- Tópico do tipo T (somente/não) pode ser usado para definir contexto de ocorrências;
- Tópico do tipo T (somente/não) pode ser usado para definir contexto de nomes;
- Tópico do tipo T (somente/não) pode ser usado para definir contexto de associações;
- Tópico do tipo T (somente/não) pode ser usado um actor em associação.

O terceiro, preocupa-se em verificar a presença de alguma característica específica em um topic map. Em certas situações, é necessário a verificação da existência de um determinado tópico desempenhando um papel de actuação específico, para determinar que o



topic map é válido de acordo com a ideia do projectista. Para isso, este tipo de restrições habilita uma interrogação ao topic map a fim de procurar o tópico ou associação desejado, a fim de considerar o topic map válido ou não; portanto, a existência de um determinado tópico ou associação pode determinar a validade do topic map em questão. Abaixo apresentam-se as regras que compõem este tipo de restrições, as quais serão exemplificadas na Subsecção 9.3.3.

#### Restrições de Existência :

- Deve existir um tópico do tipo T com o indicador de tema I;
- Deve existir um tópico do tipo T com nome N;
- Deve existir um tópico do tipo T com nome N do tipo TN;
- Deve existir um tópico do tipo T com nome N no contexto SN;
- Deve existir um tópico do tipo T com ocorrência O;
- Deve existir um tópico do tipo T com ocorrência O do tipo TO;
- Deve existir um tópico do tipo T com ocorrência O no contexto SO;
- Deve existir uma associação do tipo A no contexto S;
- Deve existir uma associação do tipo A com o papel de actuação R;
- Deve existir uma associação do tipo A com o membro T1;
- Deve existir uma associação do tipo A com o papel de actuação R sendo desempenhado pelo membro T1;

As restrições acima mencionadas – classificadas como **restrições de esquema, contextuais ou de existência** – originou-se da restrições sobre **Topic Maps** apresentadas na Secção 5.4, incluindo os acréscimos à lista de requisitos de TMCL apresentados na Secção 5.4.4.

Embora todos os itens listados acima sejam vistos como restrições, há realmente uma pequena diferença na maneira de tratar os sub-conjuntos. Portanto, o desejo de ter XTche expressando ambos tem uma influência directa no projecto da linguagem e no seu processamento. Isso será abordado nas próximas secções.

## 9.2 Estendendo XML Schema para XTche

As especificações XTche podem ser muito extensas pois em certas situações pode haver um grande número de restrições devem ser definidas; desta forma, seria muito útil definir tais restrições em uma ferramenta gráfica. Para resolver este problema, decidiu-se que a sintaxe de XTche seguiria a sintaxe de XML Schema. Isso permite ao projectista decidir a melhor maneira de editar as restrições, pois possibilita 2 tipos de edições: o estilo esquemático usando um dos editores visuais de XML Schema (onde o *output* será a respectiva descrição textual); ou o formato textual, usando então qualquer editor para descrever o documento XML.

XTche é, portanto, uma linguagem baseada em XML Schema. Assim sendo, todas as especificações XTche são instâncias de XML Schema; contudo, nem toda instância de XML Schema necessariamente é uma especificação XTche. A próxima subsecção descreve o esquema que deve ser respeitado por todas as especificações desta família.

### 9.2.1 O esquema básico de uma especificação XTche

A raiz de uma especificação XTche é, invariavelmente, o elemento `<xtche>`. Este elemento é composto por uma sequência de três sub-elementos:

- `<schema-constraints>`: permite especificar as restrições de esquema;
- `<contextual-constraints>`: permite especificar as restrições contextuais;
- `<existence-constraints>`: permite especificar as restrições de existências.

Todos esses elementos acima são opcionais, mas uma especificação terá de conter pelo menos um destes tipos de restrições. Estes elementos são compostos por uma sequência, onde cada um de seus sub-elementos representa uma restrição em particular.

O diagrama da Figura 9.1 representa o esquema básico da linguagem XTche para a definição de restrições em *Topic Maps*. A seguir ao diagrama, tem-se a versão textual desse esquema na notação XML Schema. Ele inicia com a especificação do elemento raiz, onde o *namespace* `xtche` deve ser declarado com o conteúdo `http://www.di.uminho.pt/~gepl/xtche` (linha 4). Depois, é necessário importar o esquema que especifica os elementos e atributos específicos de XTche (linha 5); este esquema está disponível em `http://www.di.uminho.pt/~gepl/xtche/xtche-schema.xsd` e será detalhado na próxima subsecção.

Finalmente, uma sequência dos elementos não-obrigatórios – `<schema-constraints>` (linha 9), `<contextual-constraints>` (linha 18) e `<existence-constraints>` (linha 27) – permitem a definição de todas as restrições necessárias para validar uma família de topic maps em particular.

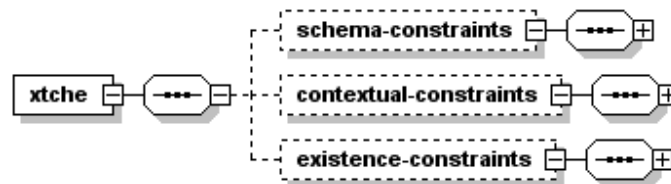


Figura 9.1: Estrutura inicial de uma especificação XTche

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
3   xmlns:xtche="http://www.di.uminho.pt/~gepl/xtche" xmlns:xs="http://www.w3.org/2001/XMLSchema">
4   <xs:import namespace="http://www.di.uminho.pt/~gepl/xtche"

```

```

5      schemaLocation="http://www.di.uminho.pt/~gepl/xtche/xtche-schema.xsd"/>
6      <xs:element name="xtche">
7          <xs:complexType>
8              <xs:sequence>
9                  <xs:element name="schema-constraints" minOccurs="0">
10                     <xs:complexType>
11                         <xs:sequence>
12                             <!-- schema constraint 1 -->
13                             ...
14                             <!-- schema constraint N -->
15                         </xs:sequence>
16                     </xs:complexType>
17                 </xs:element>
18                 <xs:element name="contextual-constraints" minOccurs="0">
19                     <xs:complexType>
20                         <xs:sequence>
21                             <!-- contextual constraint 1 -->
22                             ...
23                             <!-- contextual constraint N -->
24                         </xs:sequence>
25                     </xs:complexType>
26                 </xs:element>
27                 <xs:element name="existence-constraints" minOccurs="0">
28                     <xs:complexType>
29                         <xs:sequence>
30                             <!-- existence constraint 1 -->
31                             ...
32                             <!-- existence constraint N -->
33                         </xs:sequence>
34                     </xs:complexType>
35                 </xs:element>
36             </xs:sequence>
37         </xs:complexType>
38     </xs:element>
39 </xs:schema>

```

Onde se encontram os primeiros comentários (da linha 12 até a linha 14) no código acima, devem ser escritas as **restrições de esquema**. Caso não haja nenhuma restrição deste tipo, o elemento `<schema-constraints>` não necessita ser declarado.

O mesmo vale para os comentários das linhas 21 até a linha 23 (referentes às **restrições contextuais**) e para os comentários das linhas 30 a 32 (referentes às **restrições de existências**). Os elementos `<contextual-constraints>` e `<existence-constraints>` também são considerados dispensáveis caso não haja quaisquer restrições destes tipos.

Convém ressaltar que as especificações XTche definidas de acordo com a sintaxe XML Schema não poderão ser usadas directamente por um *parser* XML para validar um topic map em formato XTM. Isto porque, apesar de usar a referida sintaxe, o código XTche possui apenas a definição das restrições que devem ser aplicadas em Topic Maps, portanto necessita ser interpretado por um processador específico para a obtenção de um validador referente a tais restrições; o validador obtido pode garantir se o topic map a ser validado satisfaz as condições impostas. Esse processamento é descrito na Secção 9.5.

### 9.2.2 Elementos e atributos específicos da linguagem XTche

Uma vez que foi estabelecido a utilização da sintaxe de XML Schema para a definição de restrições em Topic Maps, discute-se agora como representar em XTche as restrições definidas na Secção 5.4.

Para completar o esquema básico da linguagem XTche (conforme definido na subsecção anterior) com as restrições concretas a serem impostas, recorre-se a um conjunto fixo de elementos e atributos que permitem qualificar elementos de uma especificação XTche. Esses atributos e elementos de XTche estão definidos num XML Schema específico (no ficheiro `xtche-schema.xsd` que se reproduz abaixo), o qual é importado pelo esquema básico de XTche (ver linhas 4 e 5 do esquema apresentado na subsecção anterior, Página 156).

```

1  <schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
2  xmlns:xtche="http://www.di.uminho.pt/~gepl/xtche"
3  targetNamespace="http://www.di.uminho.pt/~gepl/xtche" elementFormDefault="qualified">
4      <!-- ===== -->
5      <!-- XTche attributes for schema constraints -->
6      <!-- ===== -->
7      <attribute name="topicType"/>
8      <attribute name="subjectIndicator"/>
9      <attribute name="baseNameType"/>
10     <attribute name="baseNamePattern"/>
11     <attribute name="baseNameScope"/>
12     <attribute name="occurrenceType"/>
13     <attribute name="occurrenceScope"/>
14     <attribute name="associationType"/>
15     <attribute name="associationScope"/>
16     <attribute name="associationRole"/>
17     <attribute name="associationPlayer"/>
18     <!-- ===== -->
19     <!-- XTche attributes for contextual constraints -->
20     <!-- ===== -->
21     <attribute name="topicType-Exclusive"/>
22     <attribute name="topicType-Forbidden"/>
23     <attribute name="baseNameType-Exclusive"/>
24     <attribute name="baseNameType-Forbidden"/>
25     <attribute name="baseNameScope-Exclusive"/>
26     <attribute name="baseNameScope-Forbidden"/>
27     <attribute name="subjectIndicator-Exclusive"/>
28     <attribute name="subjectIndicator-Forbidden"/>
29     <attribute name="occurrenceType-Exclusive"/>
30     <attribute name="occurrenceType-Forbidden"/>
31     <attribute name="occurrenceScope-Exclusive"/>
32     <attribute name="occurrenceScope-Forbidden"/>
33     <attribute name="associationType-Exclusive"/>
34     <attribute name="associationType-Forbidden"/>
35     <attribute name="associationScope-Exclusive"/>
36     <attribute name="associationScope-Forbidden"/>
37     <attribute name="associationRole-Exclusive"/>
38     <attribute name="associationRole-Forbidden"/>
39     <attribute name="associationPlayer-Exclusive"/>
40     <attribute name="associationPlayer-Forbidden"/>
41     <!-- ===== -->
42     <!-- XTche attributes for existence constraints -->
43     <!-- ===== -->
44     <attribute name="topic"/>
45     <attribute name="scope"/>
46     <attribute name="baseName"/>
47     <attribute name="occurrence"/>

```

```

48     <attribute name="association"/>
49     <!-- ===== -->
50     <!-- XTche elements -->
51     <!-- ===== -->
52     <element name="baseName"/>
53     <element name="occurrence"/>
54     <element name="subjectIndicator"/>
55 </schema>

```

O XML Schema acima está separado em quatro partes.

Na primeira parte (linha 7 a 17), encontram-se os atributos que podem ser usados para a definição das **restrições de esquema**, ou seja, os atributos que qualificarão os elementos da especificação XTche, definindo a estrutura a ser seguida pelos tópicos e associações.

Como exemplo, toma-se a Figura 9.2. Essa figura apresenta a estrutura referente aos tópicos que são instâncias do tipo de tópico *artigo*. Percebe-se que o elemento *artigo* possui o atributo `@xtche:topicType`, indicando que se está a definir um tipo de tópico. Todo tópico deste tipo é então composto pela sequência de 3 elementos:

- *inglês* e *português* definem contextos de nomes (pois são qualificados com o atributo `@xtche:baseNameScope`), indicando que os tópicos do tipo *artigo* devem possuir dois nomes, um em cada contexto;
- *URL-PDF* define um tipo de ocorrência, pois ele é qualificado com o atributo `@xtche:occurrenceType`; ou seja, um tópico do tipo *artigo* tem de possuir uma ocorrência do tipo *URL-PDF*.

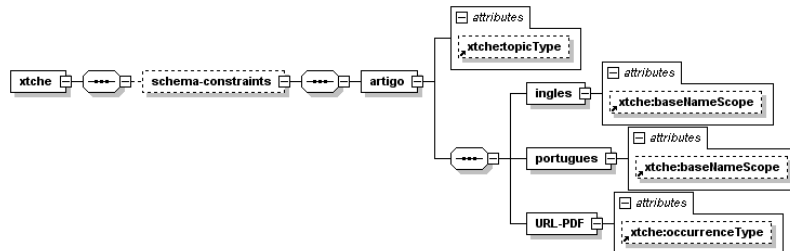


Figura 9.2: Estrutura do tipo de tópico *Artigo* com atributos XTche

Abaixo é apresentado um exemplo de tópico que obedece a restrição anterior. O tópico identificado por *XML2004-XTche* (linha 1) é uma instância do tópico *artigo* (linha 2 a 4). Além disso, possui dois nomes, onde o primeiro está no contexto *inglês* (linha 5 a 10) e o segundo, no contexto *português* (linha 11 a 16). Por fim, o tópico apresenta uma ocorrência do tipo *URL-PDF* (linha 17 a 22), fazendo uma referência ao recurso onde o tema que este tópico representa pode ser encontrado.

```

1 <topic id="XML2004-XTche">
2   <instanceOf>
3     <topicRef xlink:href="#artigo"/>
4   </instanceOf>

```

```

5      <baseName>
6          <scope>
7              <topicRef xlink:href="#ingles"/>
8          </scope>
9          <baseNameString>XTche - A Language for Topic Maps Schema and Constraints</baseNameString>
10     </baseName>
11     <baseName>
12         <scope>
13             <topicRef xlink:href="#portugues"/>
14         </scope>
15         <baseNameString>XTche - Uma Linguagem para Restrições e Esquema de Topic Maps</baseNameString>
16     </baseName>
17     <occurrence>
18         <instanceOf>
19             <topicRef xlink:href="#URL-PDF"/>
20         </instanceOf>
21         <resourceRef xlink:href="https://repositorium.sdum.uminho.pt/handle/1822/681"/>
22     </occurrence>
23 </topic>

```

Os atributos para **restrições contextuais**, definidos na segunda parte do XML Schema acima (linha 21 a 40), definem em que contexto do topic map cada elemento em particular pode estar. Esses atributos podem ser divididos em dois sub-grupos:

**com o sufixo *exclusive*:** limitam o uso de um tópico a função específica no topic map. Por exemplo: usa-se o atributo `@xtche:topicType-Exclusive` para determinar que um tópico deve ser somente usado como tipo de tópico e nada mais;

**com o sufixo *forbidden*:** proíbe o uso de um tópico em determinada função num topic map. Por exemplo: o atributo `@xtche:associationRole-Forbidden` é usado para determinar que um tópico não pode ser utilizado como um papel de actuação em associação.

A fim de exemplificar o emprego de restrições contextuais, a Figura 9.3 apresenta as regras que os tópicos *ingles*, *português* e *URL-PDF* devem obedecer. Esses tópicos são os mesmos encontrados na Figura 9.2, os quais fazem parte da estrutura dos tópicos que são instância do tipo *artigo*.

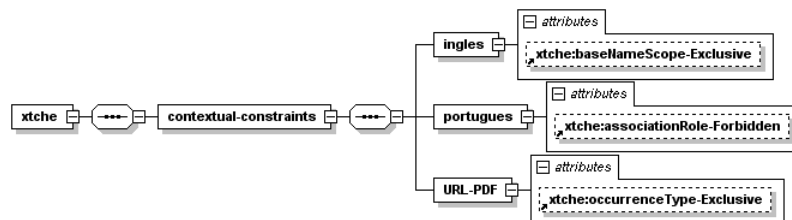


Figura 9.3: Regras contextuais para os tópicos *ingles*, *português* e *URL-PDF*

Na Figura 9.3, o elemento *ingles* possui o atributo `xtche:baseNameScope-Exclusive`, indicando que o tópico *ingles* somente poderá ser um contexto de nomes de tópicos no topic map a ser validado. Qualquer outra utilização para este tópico caracteriza um

erro. O mesmo ocorre com o elemento *URL-PDF*: por ser qualificado com o atributo `xtche:occurrenceType-Exclusive`, ele somente pode ser utilizado como um tipo de ocorrência.

Por sua vez, o elemento *português* possui o atributo `xtche:associationRole-Forbidden`; isso quer dizer que o tópico *português* pode ser encontrado em qualquer parte de um topic map, menos como um papel de actuação em associação.

A terceira parte do esquema da Página 158 contém os atributos para a especificação de **restrições de existência** (linha 44 a 48). Estes atributos são utilizados – juntamente com os atributos de **restrições de esquema** – para verificar se tópicos (ou associações) específicos existem num topic map.

A regra apresentada na Figura 9.4 verifica se o tópico *XML2004-XTche* possui um nome no contexto *inglês*. Essa leitura é obtida porque o elemento *XML2004-XTche* tem o atributo `xtche:topic` (ou seja, é visto como um tópico no topic map) e o sub-elemento *inglês* com o atributo `xtche:baseNameScope` (ou seja, é visto como um contexto de nome de um tópico).

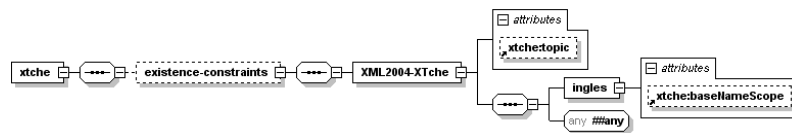


Figura 9.4: Regras para verificação de existência para o tópico *XML2004-XTche*

O elemento *any* determina que qualquer outras características de tópicos podem ser encontradas no tópico em questão, como por exemplo um nome em outro contexto, ocorrências, etc. Em outras palavras, torna a condição aberta para as demais características de tópicos que não são especificadas na regra.

A última parte do XML Schema da Página 158 apresenta três elementos (linhas 52 a 54), os quais são utilizados para determinar a cardinalidade dos indicadores de temas, dos nomes e das ocorrências de tópicos em **restrições de estrutura**. Para exemplificar, a Figura 9.5 apresenta o uso de um destes elementos para determinar que os tópicos do tipo *cliente* devem ter exactamente três ocorrências.

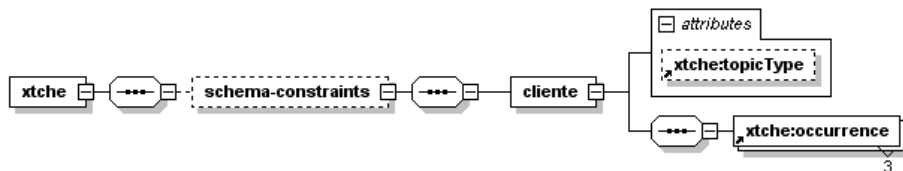


Figura 9.5: Definição do tipo de tópico *cliente* com exactas três ocorrências

De acordo com a Figura 9.5, um tópico do tipo *cliente* deve possuir exactamente 3 ocorrências para ser considerado válido. Caso contrário, será mostrado ao projectista um erro indicando quais instâncias do tópico *cliente* que não estão de acordo com esta regra.

Os dois XML Schemas (o esquema básico de XTche apresentado na Subsecção 9.2.1 e o referente aos atributos e elementos de XTche mencionado nesta subsecção) formam a base para as especificações de restrições.

### 9.3 Restringindo Topic Maps em XTche

A partir da classificação das restrições realizada na Secção 9.1, esta secção tem por objectivo apresentar como cada tipo de restrição é especificado em XTche. Composto por três subsecções (uma cada para tipo de restrição – esquema, contextuais e de existência), esta secção exemplifica as restrições listadas anteriormente.

#### 9.3.1 Especificação de Restrições de Esquema

As restrições de esquemas são aquelas regras que permitem a definição da estrutura que tópicos, ocorrências ou associações de um determinado tipo devem seguir; ou seja, possibilitam verificar se determinados objectos seguem uma estruturação previamente determinada num topic map.

O nome **restrições de esquema** vem da possibilidade de gerar esquemas de topic maps (*Topic Map Schema*<sup>2</sup>) a partir de especificações XTche.

A especificação de cada restrição de esquema será definida num sub-elemento de `<schema-constraints>` (introduzido na Secção 9.2.1). Cada elemento pode conter vários sub-elementos estruturados de acordo com as regras definidas no Apêndice A. Essas regras devem ser obedecidas pelo projectista no momento da codificação de uma especificação XTche.

Por exemplo: para especificar que todo tópico do tipo **país** deve conter um ocorrência do tipo **mapa** inserida no contexto **geografia**, deve-se escrever o código XTche abaixo:

```

1  <xs:element name="pais">
2    <xs:complexType>
3      <xs:sequence>
4        <xs:element name="mapa">
5          <xs:complexType>
6            <xs:sequence>
7              <xs:element name="geografia">
8                <xs:complexType>
9                  <xs:attribute ref="xtche:occurrenceScope"/>
10               </xs:complexType>
11             </xs:element>
12           </xs:sequence>
13           <xs:attribute ref="xtche:occurrenceType"/>
14         </xs:complexType>
15       </xs:element>
16     </xs:sequence>
17     <xs:attribute ref="xtche:topicType"/>
18   </xs:complexType>
19 </xs:element>

```

<sup>2</sup> *Topic Map Schema* é descrito em detalhe na Secção 9.4.



Em vez de codificar textualmente (como realizado acima), a mesma especificação pode ser graficamente definida num editor de XML Schema, conforme mostrado na Figura 9.6, a qual é o diagrama respectivo ao código anterior.

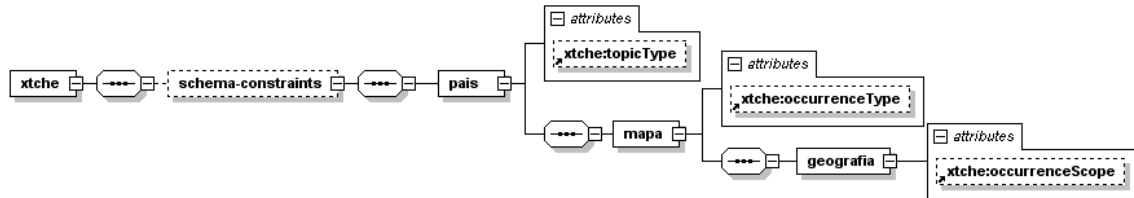


Figura 9.6: Uma especificação XTche para definir a estrutura do tópico *country*

Como mostrado na Figura 9.6, uma restrição de esquema é uma sequência de tópicos concretos (*país*, *mapa* e *geografia*), cada um qualificado com um atributo XTche específico (@xtche:topicType, @xtche:occurrenceType e @xtche:occurrenceScope, respectivamente).

A partir de uma restrição de esquema, um esqueleto XTM pode ser gerado automaticamente, definindo a forma que os tópicos, associações e ocorrências devem assumir no topic map, definindo as suas características. Isso será detalhado na Secção 9.4.

Um exemplo de aplicação de XTche mais sofisticado é inspirado na Secção 6.1 de *TMCL Requirements* (Nishikawa and Moore, 2003b) (visto em detalhe no Capítulo 13). Este caso trata de uma aplicação electrónica comercial (*E-Commerce Application*). Neste exemplo, um dos relacionamentos é definido pelo tipo de associação *comprar*, o qual possui dois papéis de actuação: *cliente* e *compra*. O papel *compra* deve ser desempenhado por, ao menos, um tópico do tipo *compra*, e o papel *cliente* desempenhado por um actor, o qual deve ser um tópico do tipo *cliente* ou *empregado*<sup>3</sup>. Esta especificação XTche acima pode ser representada pelo diagrama apresentado na Figura 9.7.

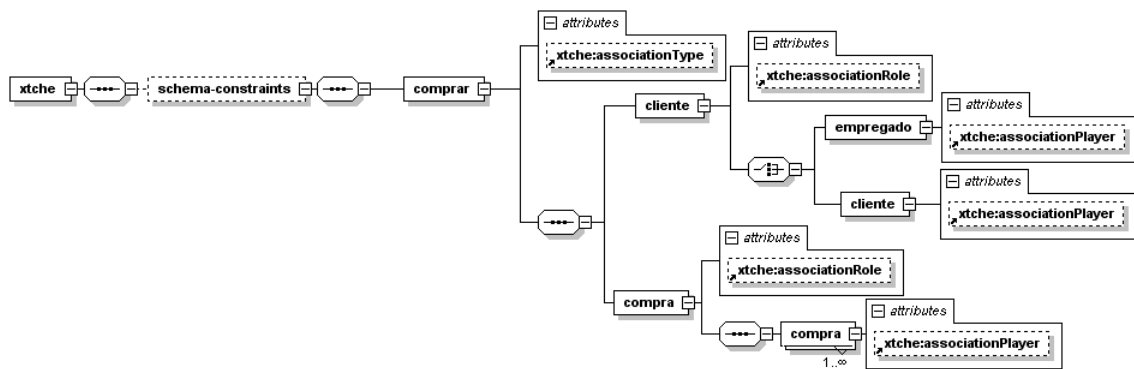


Figura 9.7: Especificação XTche para definir a estrutura da associação *comprar*

O código XTche abaixo reflecte a restrição definida na Figura 9.7:

<sup>3</sup>O membro *empregado* foi aqui incluído para apresentar a especificação de alternativas em XTche, não existindo no caso de estudo proposto nos requisitos de TMCL.

```

1  <xs:element name="comprar">
2    <xs:complexType>
3      <xs:sequence>
4        <xs:element name="cliente">
5          <xs:complexType>
6            <xs:choice>
7              <xs:element name="empregado">
8                <xs:complexType>
9                  <xs:attribute ref="xtche:associationPlayer"/>
10               </xs:complexType>
11             </xs:element>
12             <xs:element name="cliente">
13               <xs:complexType>
14                 <xs:attribute ref="xtche:associationPlayer"/>
15               </xs:complexType>
16             </xs:element>
17           </xs:choice>
18           <xs:attribute ref="xtche:associationRole"/>
19         </xs:complexType>
20       </xs:element>
21       <xs:element name="compra">
22         <xs:complexType>
23           <xs:sequence>
24             <xs:element name="compra" maxOccurs="unbounded">
25               <xs:complexType>
26                 <xs:attribute ref="xtche:associationPlayer"/>
27               </xs:complexType>
28             </xs:element>
29           </xs:sequence>
30           <xs:attribute ref="xtche:associationRole"/>
31         </xs:complexType>
32       </xs:element>
33     </xs:sequence>
34     <xs:attribute ref="xtche:associationType"/>
35   </xs:complexType>
36 </xs:element>

```

O elemento `<xs:choice>` (linha 6 a 17) define os dois actores alternativos que podem ser encontrados desempenhando o papel *cliente* no topic map: ou *empregado* (linha 7) ou *cliente* (linha 12). O atributo *maxOccurs*, associado ao actor *compra* (linha 24), define a cardinalidade (neste caso, um ou mais) de actores que podem desempenhar o papel *compra*.

A partir disso, entende-se que toda associação do tipo *comprar* possui um tópico do tipo *cliente* ou do tipo *empregado* desempenhando o papel *cliente*, enquanto que um ou mais tópicos do tipo *compra* desempenharão o papel *compra*.

### 9.3.2 Especificação de Restrições Contextuais

As restrições contextuais são especificadas em XTche como sub-elementos do elemento `<contextual-constraints>`, como comentado na Secção 9.2.1.

Cada sub-elemento definido como uma restrição contextual será referente à um tópico em específico. Isto significa que tais restrições são adoptadas para determinar qual a aplicabilidade de um tópico em particular numa família de topic maps. Ou seja, define-se em que situações um determinado tópico pode (ou não pode) ser encontrado no topic map.

Os elementos referentes às restrições contextuais não possuem sub-elementos; apenas atributos para os qualificarem de acordo com a intenção do autor da especificação.

Por exemplo, para declarar que o tópico *perfil* *pode ser usado para definir tipos de ocorrências e nada mais*, tudo o que deve ser feito é adicionar o atributo `@xtche:occurrenceType-Exclusive` ao elemento *perfil*, como visualizado na Figura 9.8 e mostrado no código XTche a seguir:

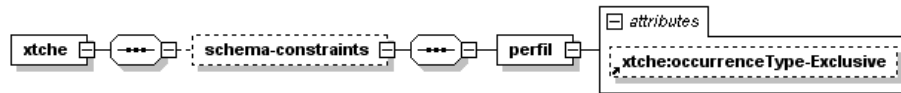


Figura 9.8: Restrição contextual determinando que o tópico *perfil* é, exclusivamente, um tipo de ocorrência

```

1  <xs:element name="contextual-constraints">
2    <xs:complexType>
3      <xs:sequence>
4        <xs:element name="perfil">
5          <xs:complexType>
6            <xs:attribute ref="xtche:occurrenceType-Exclusive"/>
7          </xs:complexType>
8        </xs:element>
9      </xs:sequence>
10   </xs:complexType>
11 </xs:element>

```

O nome desta família de restrições justifica-se porque as mesmas são aplicadas sobre o contexto do objecto que está sofrendo a restrição em questão (no caso, um tópico que somente pode ser um tipo de ocorrências): ela refere-se somente ao próprio objecto, não se preocupando com sua estrutura.

Em termos práticos, para validar as declarações de restrições acima, o TM-Validator necessita verificar se o tópico *perfil* somente está sendo referenciado no caminho XPath `//occurrence/instanceOf/topicRef/@xlink:href`. Caso este tópico seja encontrado em algum outro sítio do topic map, uma mensagem de erro informará tal situação e o topic map não será considerado válido.

Outro exemplo consiste em confirmar que o tópico *artigo* *não pode ser usado como tipo de tópicos e como tipo de associações*. Na linguagem XTche, simplesmente é necessário adicionar o elemento `<artigo>` na secção de *restrições contextuais*, com os atributos `@xtche:topicType-Forbidden` e `@xtche:associationType-Forbidden`, conforme apresentado visualmente na Figura 9.9 e textualmente no código XTche a seguir. Assim, toda vez que o TM-Validator encontrar o tópico *artigo* sendo utilizado como tipo de tópico ou como tipo de associação, erros serão reportados.

```

1  <xs:element name="contextual-constraints">
2    <xs:complexType>
3      <xs:sequence>
4        <xs:element name="artigo">
5          <xs:complexType>
6            <xs:attribute ref="xtche:topicType-Forbidden"/>
7            <xs:attribute ref="xtche:associationType-Forbidden"/>

```

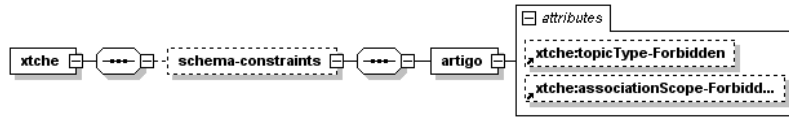


Figura 9.9: Restrição contextual determinando que o tópico `artigo` não pode ser tipo de tópico, nem contexto de associação

```

8      </xs:complexType>
9      </xs:element>
10     </xs:sequence>
11  </xs:complexType>
12 </xs:element>

```

De acordo com a especificação XTche acima, percebe-se que o tópico `artigo` não deve ser encontrado, num topic map na sintaxe XTM, nos caminhos XPath `//topic/instanceOf/topicRef` (referente aos tipos dos tópicos) e `//association/instanceOf/topicRef` (referente aos tipos de associações).

Assim, com o auxílio da linguagem XTche, o autor pode facilmente verificar em quais segmentos do código XTM há erros, para então solucioná-los.

### 9.3.3 Especificação de Restrições de Existência

As **restrições de existência** foram projectadas para a avaliação de topic maps perante confirmação da presença (existência) de tópicos, associações ou ocorrências específicos com certas características. Portanto, essas restrições não se preocupam com um conjunto de tópicos ou associações de um mesmo tipo; seu processo consiste de verificar se o elemento procurado existe ou não existe exactamente da forma como se deseja.

Se *re* é uma **restrição de existência**, então um topic map *tm* será considerado válido se, ao menos, um tópico *t* ou associação *a* está de acordo com *re*. A Figura 9.10 apresenta a restrição onde será considerado válido o topic map se ele possuir uma associação do tipo *orientar* (atributo `@xtche:associationType`) que possua o papel *orientador* (atributo `@xtche:associationRole`) sendo desempenhado pelo tópico *prh* (atributo `@xtche:topic`).

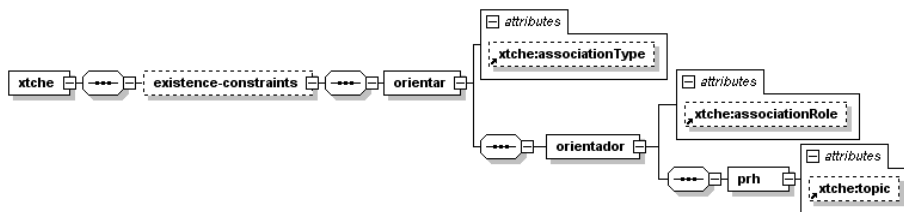


Figura 9.10: Exemplo de especificação de restrição de existência rígida

Para um topic map ser considerado válido, ele deve conter ao menos uma associação que possua exclusivamente o papel *orientador* sendo desempenhado pelo tópico referente

a *Pedro Henriques*. Quaisquer outros papéis ou tópicos membros na mesma associação indicam que ela não está de acordo com a regra.

Caso se deseja procurar se há alguma associação do tipo *orientar* conforme a citada acima, porém sem a restrição a demais papéis de actuação e membros, é necessário a declaração do elemento **any**, conforme demonstrado na Figura 9.11.

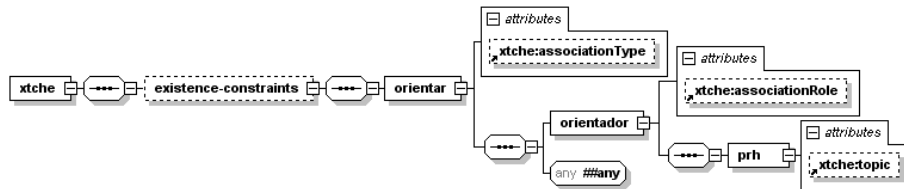


Figura 9.11: Exemplo de especificação de restrição de existência não-rígida

O elemento **any** define que quaisquer outros papéis de actuação e membros podem ser encontrados nessa associação. O TM-Validator se preocupará em encontrar uma associação que tenha o papel *orientador* sendo desempenhado pelo tópico *prh*, independente se a mesma associação tem, ou não, outros papéis e membros. Ao encontrar ao menos uma associação que esteja de acordo com a regra definida na Figura 9.11, o topic map será considerado válido.

## 9.4 Estruturando Topic Maps com XTche

Ao analisar as **restrições de esquema** apresentadas anteriormente, se percebe claramente que pode-se definir a estrutura completa dos tópicos e associações de uma família de topic maps, ou seja, é possível definir um *Topic Map Schema* a partir de uma especificação XTche.

Com um *Topic Map Schema* pode-se obter um modelo que os tópicos e associações de um determinado tipo devem seguir. Este modelo pode ser gerado automaticamente pelo Gerador de Esqueleto XTM conforme apresenta a Figura 9.12; tomando como entrada a especificação XTche, processador gera como saída um código XTM, chamado de *esqueleto de topic maps*.



Figura 9.12: Gerador de Esqueleto XTM a partir de especificações XTche

Analogicamente, pode-se comparar a relação entre um topic map e um *Topic Map Schema* com a relação entre XML e XML-Schemas. O XML Schema define as regras que as

instâncias XML devem seguir para serem consideradas válidas. O *Topic Map Schema* define uma colecção de restrições que pode ser usada para estruturar cada objecto de um topic map instância. Assim, todos os topic maps que estão de acordo com um *Topic Map Schema* pertencem a mesma família.

Um ponto importante sobre esquemas é que eles podem ser vistos como a documentação para as instâncias baseadas no próprio esquema, ou seja, o esquema define qual a estrutura dos tópicos e associações de um domínio. O esquema especifica o necessário e indispensável para a criação de topic maps referentes a um certo domínio.

Um *Topic Map Schema* permite certas funcionalidades, tais como (Nishikawa, Moore, and Bogachev, 2004):

- A possibilidade de geração automática de interfaces para auxiliar os editores na criação de topic maps válidos. A interface pode, alternativamente, ser menos restritiva e apenas notificar quando o editor está fazendo algo considerado inválido pela estrutura do topic map em questão;
- Uma interface para utilizador que seja apta a fazer inferências sobre informação presente no topic map e apresenta-las para o editor quando isto for solicitado;
- A diminuição do tempo de criação de topic maps concretos, pela ajuda extra que o editor recebe da ferramenta.

Um esquema de topic map define qual será a ontologia a ser representada no topic map, de acordo com o apresentado na Secção 4.3. Em outras palavras, o esquema irá determinar qual será a estrutura que os tópicos, as associações e as ocorrências deverão seguir para estes serem considerados correctos.

Além disto, é possível inferir um esqueleto de topic map (escrito em XTM) a partir do esquema XTche; o utilizador ou uma aplicação (como o Oveia – Capítulo 8) deve somente preenchê-lo (com dados extraídos dos recursos de informação) para obter as instâncias de topic maps.

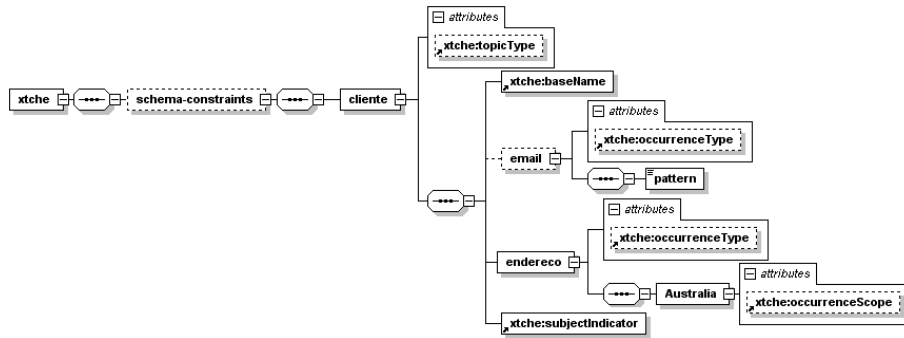
A definição de um esquema para um tipo de tópico fornece todos os construtores que permitem efectuar a geração de um esqueleto deste tipo, o qual pode ser utilizado por qualquer instância sua.

Toma-se o tipo de tópico *cliente* como exemplo: ele é composto por um nome, uma ocorrência (do tipo *endereço* inserida no contexto *Austrália*), um identificador de tema (para o identificador do cliente) e, opcionalmente, uma ocorrência adicional para o *e-mail* (deve conter o caracter “@”). Seu esquema está representado na Figura 9.13 e é descrito a seguir.

```

1 <xs:element name="cliente">
2   <xs:complexType>
3     <xs:sequence>
4       <xs:element ref="xtche:baseName"/>
5       <xs:element name="email" minOccurs="0">
6         <xs:complexType>

```

Figura 9.13: Especificação XTche para o tipo de tópico *cliente*

```

7      <xs:sequence>
8          <xs:element name="pattern">
9              <xs:simpleType>
10                 <xs:restriction base="xs:string">
11                     <xs:pattern value="@"/>
12                 </xs:restriction>
13             </xs:simpleType>
14         </xs:element>
15     </xs:sequence>
16     <xs:attribute ref="xtche:occurrenceType"/>
17 </xs:complexType>
18 </xs:element>
19 <xs:element name="endereco">
20     <xs:complexType>
21         <xs:sequence>
22             <xs:element name="Australia">
23                 <xs:complexType>
24                     <xs:attribute ref="xtche:occurrenceScope"/>
25                 </xs:complexType>
26             </xs:element>
27         </xs:sequence>
28         <xs:attribute ref="xtche:occurrenceType"/>
29     </xs:complexType>
30 </xs:element>
31 <xs:element ref="xtche:subjectIndicator"/>
32 </xs:sequence>
33 <xs:attribute ref="xtche:topicType"/>
34 </xs:complexType>
35 </xs:element>

```

Tomando esta definição por base, o Gerador de Esqueleto XTM realiza a criação de um esqueleto de tópico do tipo *cliente* na sintaxe XTM com as características definidas em XTche. A partir deste esqueleto, o utilizador pode preencher cada instância deste com os dados que bem entender, seja manualmente, seja através de uma aplicação de extracção automática. Assim, o código XTM gerado é similar ao abaixo apresentado:

```

1 <topic id="cliente"/>
2 <topic id="endereco"/>
3 <topic id="Australia"/>
4 <topic id="email"/>
5 <topic id="CLIENTE001">
6     <instanceOf>
7         <topicRef xlink:href="#cliente"/>

```

```

8 | </instanceOf>
9 | <subjectIdentity>
10 |   <subjectIndicatorRef xlink:href="http://SUBJECT001"/>
11 | </subjectIdentity>
12 | <baseName>
13 |   <baseNameString>BASENAME001</baseNameString>
14 | </baseName>
15 | <occurrence>
16 |   <instanceOf>
17 |     <topicRef xlink:href="#endereco"/>
18 |   </instanceOf>
19 |   <scope>
20 |     <topicRef xlink:href="#Australia"/>
21 |   </scope>
22 |   <resourceData>RESOURCE001</resourceData>
23 | </occurrence>
24 | <!--occurrence>
25 |   <instanceOf>
26 |     <topicRef xlink:href="#email"/>
27 |   </instanceOf>
28 |   <resourceRef xlink:href="RESOURCE002"/>
29 | </occurrence-->
30 | </topic>

```

O esqueleto acima define 5 tópicos: os 4 primeiros (linha 1 a 4) são os tópicos que são especificados como tipo de tópicos (*cliente*), tipos de ocorrências (*endereço* e *email*) e contexto de ocorrências (*Australia*).

O quinto tópico (linha 5 a 39) define a estrutura de todo tópico que é uma instância de *cliente*. Os sítios que estão preenchidos por conteúdos em caracteres maiúsculos seguidos de 3 dígitos (linhas 5, 10, 13, 22 e 28) devem então ser substituídos por informações referentes a cada tópico em questão.

A ocorrência do tipo *email* (linhas 24 a 29) está entre comentários porque ela foi definida como opcional no esquema deste tipo de tópico. Assim, quando ela deve ser mencionada, os comentários são retirados e preenche-se a informação devida no elemento **resourceRef**. Quando esta característica de um tópico em particular deste tipo não existe, basta manter os comentários.

Para uma associação, o processo é o mesmo. A partir de uma especificação de um esquema de um tipo de associação pode-se obter um esqueleto contendo as características definidas nas regras impostas.

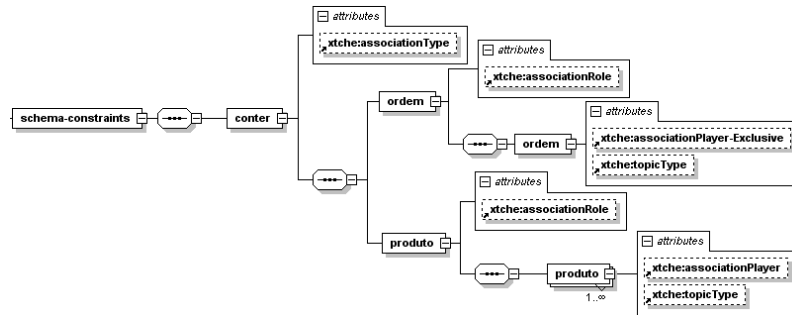
Para exemplificar este caso, usa-se a associação do tipo *conter* que é composta por um membro do tipo *ordem* desempenhando o papel de actuação *ordem* e por um ou mais membros do tipo *produto* desempenhando o papel *produto*. A especificação XTche referente a este tipo de associação apresenta-se na Figura 9.14 e é descrita logo a seguir:

```

1 | <xs:element name="conter">
2 |   <xs:complexType>
3 |     <xs:sequence>
4 |       <xs:element name="ordem">
5 |         <xs:complexType>
6 |           <xs:sequence>
7 |             <xs:element name="ordem">
8 |               <xs:complexType>

```



Figura 9.14: Especificação XTche para o tipo de associação *conter*

```

9      <xs:attribute ref="xtche:associationPlayer"/>
10     <xs:attribute ref="xtche:topicType"/>
11   </xs:complexType>
12 </xs:element>
13 </xs:sequence>
14   <xs:attribute ref="xtche:associationRole"/>
15 </xs:complexType>
16 </xs:element>
17 <xs:element name="produto">
18   <xs:complexType>
19     <xs:sequence>
20       <xs:element name="produto" maxOccurs="unbounded">
21         <xs:complexType>
22           <xs:attribute ref="xtche:associationPlayer"/>
23           <xs:attribute ref="xtche:topicType"/>
24         </xs:complexType>
25       </xs:element>
26     </xs:sequence>
27     <xs:attribute ref="xtche:associationRole"/>
28   </xs:complexType>
29 </xs:element>
30 </xs:sequence>
31   <xs:attribute ref="xtche:associationType"/>
32 </xs:complexType>
33 </xs:element>

```

Baseado na especificação XTche da Figura 9.14, o Gerador de Esqueleto XTM determina a criação do seguinte esqueleto de associação:

```

1  <association>
2    <instanceOf>
3      <topicRef xlink:href="#conter"/>
4    </instanceOf>
5    <member>
6      <roleSpec>
7        <topicRef xlink:href="#ordem"/>
8      </roleSpec>
9      <topicRef xlink:href="#MEMBER001"/>
10    </member>
11    <member>
12      <roleSpec>
13        <topicRef xlink:href="#produto"/>
14      </roleSpec>
15      <topicRef xlink:href="#MEMBER002"/>
16    </member>

```

```
17      <!--member>
18          <roleSpec>
19              <topicRef xlink:href="#produto"/>
20          </roleSpec>
21          <topicRef xlink:href="#MEMBER003"/>
22      </member-->
23  </association>
```

Com isso, entende-se que toda associação do tipo *conter* será composta por no mínimo dois membros, onde o primeiro desempenha o papel *ordem* (linha 5 a 10) enquanto que o segundo desempenha o papel *produto* (linha 11 a 16). Além disso, outros membros desempenhando o papel de actuação *produto* podem se fazer presentes neste tipo de associação; por isto, o terceiro membro (linha 17 a 22) está envolvido por comentários.

## 9.5 Processando XTche

As especificações escritas na linguagem XTche, referentes a um conjunto de restrições – listando as condições (envolvendo tópicos e associações) que devem ser verificadas – especificam um processo de validação em topic maps (um TM-Validator).

Aliado ao facto de XTche ser uma linguagem XML e ao facto de que o formato escolhido para descrever o topic map a validar foi XTM, foi possível fazer a codificação sistemática, em XSL, desta tarefa de verificação. A partir destas circunstâncias, tornou-se possível gerar automaticamente este validador.

Para este propósito, desenvolveu-se o Processador XTche (uma folha de estilos XSL) que efectua a criação de um TM-Validator (também uma folha de estilos XSL) a partir de uma especificação XTche. Ambas folhas de estilo XSL (o gerador e o validador) são interpretados por um processador XSL padrão, como o Saxon<sup>4</sup>, o que é um dos benefícios desta proposta. Em relação a isso, a abordagem XTche à validação se assemelha à abordagem TM-Builder à construção, no sentido que ambos são totalmente baseados em XML.

Funcionando nos mesmos moldes de Schematron (Dodds, 2001) e XSCL (Jacinto et al., 2003), as regras XTche determinarão a construção de uma folha de estilos XSL – TM-Validator – que será responsável por avaliar o topic map. O processamento de tal folha de estilos actuará como um verificador automático que garantirá que o topic map está, ou não, de acordo com as restrições previamente definidas.

O Processador XTche é o gerador de TM-Validator; ele comporta-se precisamente como um gerador de compilador e é o núcleo da arquitectura de validação semântica de Topic Maps, como pode ser visto na Figura 9.15. Ele toma uma especificação XTche e gera o TM-Validator que irá processar o topic map para validá-lo e gerar mensagens de erros, à medida que os mesmos forem sendo detectados.

---

<sup>4</sup>Saxon – <http://saxon.sourceforge.net/>

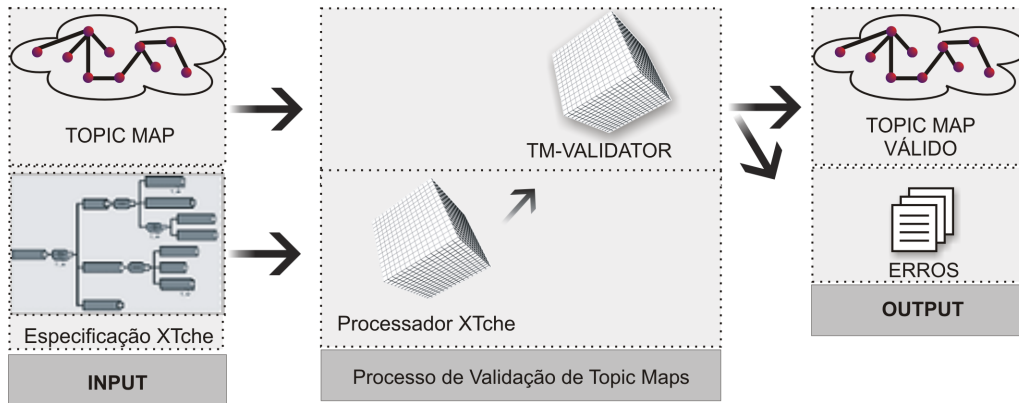


Figura 9.15: Arquitectura de XTche

### 9.5.1 TM-Validator

Para conferir a validade de uma restrição XTche num topic map deve-se seguir um conjunto de regras. Esse conjunto de regras é o algoritmo seguido pelo TM-Validator.

Se  $r$  é uma restrição a ser aplicada, então  $r$  estará de acordo com o conjunto de tópicos  $t$  ou o conjunto de associações  $a$  se ele satisfizer as seguintes regras:

- para todos os tópicos em  $t$ :
  1. o identificador de tópico em  $r$  e  $t$  estão de acordo;
  2. todos os tipos de tópicos em  $r$  estão de acordo com os de  $t$ ;
  3. todas as características de tópico expressas em  $r$  existem em  $t$  e estão de acordo;
- para todas as associações em  $a$ :
  1. o tipo de associação em  $r$  e  $a$  estão de acordo;
  2. os membros e papéis de actuação em associações em  $r$  existem em  $a$  e estão de acordo.

A partir deste algoritmo básico, o TM-Validator encarrega-se de verificar restrição por restrição, conferindo se há de facto uma concordância entre as regras definidas pelo utilizador e o topic map a ser validado.

### Problemas e soluções no desenvolvimento do TM-Validator

Alguns problemas foram encontrados durante o desenvolvimento do TM-Validator tiveram um grande impacto no seu algoritmo de validação. O mais importante deles foi a ambiguidade na definição da forma como as restrições devem ser aplicadas aos topic maps. Como dito anteriormente, uma especificação XTche é composta de um conjunto de restrições,

mas não foi dito que estas restrições são disjuntas em termos de contexto; ou seja, cada restrição é interpretada independentemente das demais.

Em alguns casos, há uma certa sobreposição entre os contextos de diferentes restrições; esta sobreposição pode causar um erro quando codificado em XSL, pois processadores XSL somente podem verificar a combinação de um contexto por vez. A solução adoptada para superar este problema foi executar cada restrição em um *mode*<sup>5</sup> diferente. Desta forma, cada restrição é tratada de forma independente, evitando ambiguidades entre as mesmas.

Em termos de optimização das consultas a serem efectuadas em topic maps, usa-se uma potencialidade de XSL para a criação de *tabelas Hash* através dos elementos `xsl:key` (Holman, 2001). Este elemento permite criar um índice para um determinado grupo de elementos (especificado por uma expressão XPath). Assim, para cada conjunto de elementos do topic map (tópicos, associações, ocorrências, nomes, etc.) será criado um índice baseado nas referências encontradas no atributo `@xlink:href`. Esse índice optimiza a consulta, evitando que o topic map seja percorrido elemento por elemento, para a avaliação de cada restrição. O ganho em termos temporais é exponencial, pois só se percorre o topic map uma vez, para a formação do índice.

### 9.5.2 Validação de Especificações XTche

Como dito anteriormente, uma especificação XTche obrigatoriamente deve ser um documento XML Schema; contudo, nem todo XML Schema é uma especificação XTche. Assim, um *parser* XML Schema não é um mecanismo suficiente para confirmar que uma especificação pertence à linguagem XTche pois é necessário uma garantia que o documento XML Schema pertença à classe de especificações XTche.

Deste modo, a arquitectura foi estendida em mais uma camada – chamada XTche-SpecVP – responsável pela validação das especificações XTche. A Figura 9.16 descreve o processo XTche-SpecVP – formado pelo par de processadores Validador de Schema e Validador de XTche – cujo funcionamento é descrito abaixo:

1. primeiramente, verifica-se a especificação XTche é uma instância XML Schema válida através de um Validador de Schema, ou seja, um *parser* XML Schema comum, que pode estar incorporado a uma ferramenta integrada de desenvolvimento – como o XMLSpy® – ou pode ser uma ferramenta específica para este fim – como o *XML Schema Quality Checker* (Fokoué et al., 2001).
2. se a especificação XTche é um XML Schema, o Validador de XTche verifica a obediência da referida especificação de acordo com as regras definidas no Apêndice A, as quais completam o esquema básico da linguagem XTche, apresentado na Secção 9.2.1;
3. os erros são reportados conforme eles ocorram. Portanto, a especificação XTche é considerada correta somente se nenhum erro for detectado.

---

<sup>5</sup>Em XSL, cada *mode* corresponde a uma diferente travessia na árvore XML.

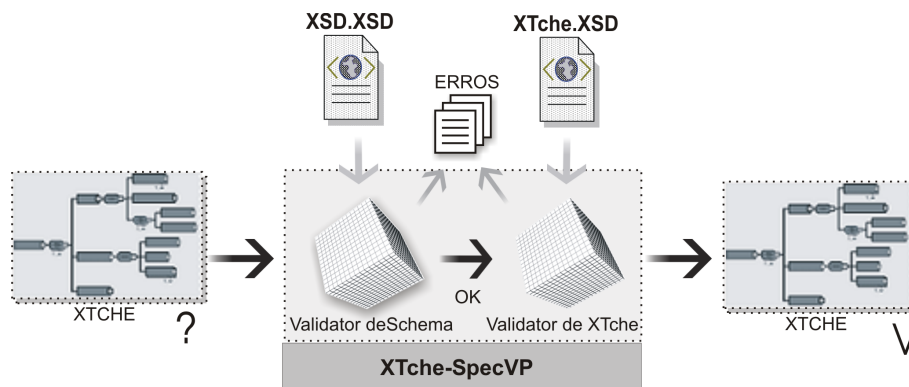


Figura 9.16: Processador para a Validação de Especificações XTche

O Validador de XTche implementa um conjunto de regras, as quais devem ser cumpridas pela especificações XTche. Tais regras estão divididas em 3 sub-conjuntos: regras para **restrições em esquemas**, **contextuais** e **de existência**, as quais estão listadas no Apêndice A.

Se uma especificação XTche não desrespeitar nenhuma das regras descritas, a especificação será considerada válida. Contudo, se ao menos uma regra for desobedecida, os erros serão reportados ao utilizador e a validação da especificação não será confirmada.

## 9.6 Trabalhos Relacionados com XTche

AsTMa! (Barta, 2003a) é uma linguagem para a especificação de restrições sobre **Topic Maps**. Robert Barta, seu criador, também propõe um mecanismo para validar um topic map de acordo com um conjunto de regras. Esta linguagem utiliza **AsTMa=** (Barta, 2004), uma linguagem de composição (*authoring language*), estende-a com vários construtores, como por exemplo os operadores lógicos (*not*, *and* e *or*), e as palavras-chave *forall* (para a selecção de sub-mapas) e *exists* (para as restrições condicionais). **AsTMa!** cobre alguns dos requisitos de TMCL. Detalhes sobre **AsTMa!** são apresentados na Subsecção B.1.

OSL (*Ontopia Schema Language*) (Garshol, 2004c) permite a definição de esquema para **Topic Maps**, restringindo a estrutura das suas instâncias. A linguagem OSL será detalhada na Subsecção B.2.

Em outro trabalho relacionado, Eric Freese (Freese, 2002) apresenta a possibilidade de se utilizar a linguagem DAML+OIL para fornecer mecanismos para restringir e validar informações em topic maps. Este artigo apresenta como descrever a validação e a consistência da informação contida em **Topic Maps** usando DAML+OIL, apresentando exemplos sobre PSI, hierarquia de classes e designação de propriedades para tópicos. Contudo, esta proposta não apresenta um processador para tais validações.

Quando é realizada uma comparação entre XTche e os trabalhos relacionados, algumas vantagens de XTche são detectadas; por exemplo, XTche é uma linguagem baseada em

XML Schema, uma recomendação da *World Wide Web Consortium* (W3C) (W3C, 2005). Aliado a isto, XTche permite a edição de restrições em um editor gráfico de XML Schema, como o XMLSpy®. Com a visão gráfica da especificação, torna-se acessível uma verificação visual sobre a correcção da mesma.

Em outro contexto, XTche e AsTma! possuem mais mecanismos para a verificação da validade de *Topic Maps*, permitindo a validação de um maior número de restrições, quando comparado com a linguagem OSL e a proposta de Eric Freese. Isto é comprovado no Apêndice B.

## 9.7 Sumário sobre XTche

O presente capítulo apresentou um sistema para a validação de *Topic Maps* composto pela linguagem XTche e seu processador. Essa ideia partiu da motivação para a verificação da correcção sintáctica e semântica de *Topic Maps* que foi discutida no Capítulo 5, pois uma notação para a descrição de uma ontologia que suporta um sistema de computação sofisticado (como por exemplo, aplicações na área de *Semantic Web*) deve ser validada.

Então, foi adoptado o formato de intercâmbio de *Topic Maps* – XTM (Pepper and Moore, 2001a) – e os requisitos para uma linguagem de restrições em *Topic Maps* – *TMCL Requirements* (Nishikawa, Moore, and Bogachev, 2004) – como pontos de partida. Além disso, foi utilizado nosso conhecimento na área de compiladores e validação semântica de documentos XML para propor esta ideia.

XTche abrange os requisitos impostos pela ISO – na definição de TMCL – apoiado numa sintaxe baseada em XML Schema.

Dois casos de estudos – uma aplicação de *E-Commerce* (apresentado no Capítulo 13) e o Museu do Emigrante (apresentado no Capítulo 14) – comprovaram esta abordagem com sucesso, representando vários dos possíveis casos cobertos por XTche.

O sistema proposto neste capítulo – XTche – comporta-se como Schematron e XCSL. Isto quer dizer que o validador semântico de *Topic Maps* (gerado de acordo com uma especificação) confere a coerência de um topic map de acordo com um conjunto de regras; após o processamento de cada topic map, mantém-se silêncio quando as restrições são satisfeitas, e detecta-se/reporta-se os erros, quando as condições são quebradas.

Há, actualmente, algumas linguagem com objectivos similares, definidas para a validação de *Topic Maps*. Contudo, estas linguagens ainda não atingiram o nível desejado pela comunidade de *Topic Maps*.

Uma das inovações dessa proposta é que a linguagem permite a definição de estrutura para topic maps em um estilo baseado em XML Schema; portanto, não é mais necessário uma descrição sintáctica em separado. Uma especificação XTche une o esquema (definição da estrutura e da semântica básica) com restrições (descrevendo a semântica contextual) para todos os topic maps de uma família em particular.

O uso da sintaxe XML Schema justifica-se, basicamente, por possibilitar uma edição e

---

visualização gráfica da especificação XTche (pois há um vasto conjunto de ferramentas que possibilitam a edição gráfica de XML Schema) e por possuir uma sintaxe bem conhecida pela comunidade XML, fornecendo assim uma interface gráfica e um verificador sintáctico básico (primeira etapa do *XTche-SpecVP*).





## Capítulo 10

# Ulisses

*Trago dentro do meu coração como num cofre que não pode fechar de cheio,  
todos os lugares onde estive, todos os portos que cheguei,  
todas as paisagens que vi através de janelas ou vigias ou de tombadilho,  
sonhando e tudo isso que é tanto é pouco para o que eu quero*

Fernando Pessoa

Recordando que os **Topic Maps** são grafos compostos por tópicos (onde cada tópico representa um tema, identifica recursos e está associado com outros tópicos), o **Ulisses** foi imaginado e desenvolvido para providenciar navegadores estáticos que permitem percorrer estas redes de conceitos. Esses navegadores são compostos por páginas HTML que descrevem os tópicos e usam a ideia de hiper-ligações para implementar as associações e as ligações às ocorrências.

Na medida em que a plataforma **Metamorphosis** (Capítulo 6) gera **Topic Maps** para organizar a informação dispersa por uma ou mais fontes, de acordo com uma dada ontologia, é elementar pensar-se em incluir esta ferramenta como um módulo final que possibilite a navegação conceptual sobre os recursos de informação tratados.

A ideia de navegação conceptual reflecte a forma em que a mente humana pensa, *baseada em informações associadas*. Por exemplo, quando se pensa no *Brasil*, automaticamente pode-se pensar:

- é um país que está situado na América do Sul (associação entre *país* e *continente*, no contexto *geografia*);
- possui mais de 170 milhões de habitantes (ocorrência do tipo *população*);
- é governado pelo presidente *Luís Inácio Lula da Silva* (associação entre *país* e *presidente*, no contexto *política*);
- é o actual campeão do mundo de futebol (associação entre *país* e *futebol*, no contexto *desporto*);

- é um país com um total de 121.391.630 eleitores (ocorrência do tipo *eleitores*).

Assim, o que se pretende neste capítulo é apresentar essa ferramenta baseada em conceitos de navegação em grafos, chamada *Ulisses*.

No contexto do *Metamorphosis*, o *Ulisses* possui a finalidade de criar interfaces web a partir dos topic maps gerados e validados pelos outros módulos desta arquitectura, respectivamente *TM-Builder/Oveia* e *XTche* (conforme apresentado na Figura 6.2).

Como se percebe do que foi dito acima, o *Ulisses* não se limita ao contexto do *Metamorphosis*; isto significa que ele possui a capacidade de gerar interfaces web para todo e qualquer topic map, que segue a sintaxe XML Topic Maps (XTM) ou que esteja armazenado no modelo relacional BD Ontologia, definido na Secção 8.6.

Para ilustrar as ideias aqui introduzidas, este capítulo está dividido em 4 partes. Na Secção 10.1, apresenta-se uma técnica de visualização de grafos e topic maps. A seguir, o *Ulisses* é descrito na Secção 10.2. As vantagens e desvantagens do *Ulisses* em relação aos trabalhos relacionados encontram-se na Secção 10.3. Por fim, um sumário encerra o capítulo com as considerações finais.

## 10.1 Visualização de Topic Maps

Antes de apresentar o *Ulisses*, esta secção introduz um modo de visualização de Topic Maps, que é o método seguido pelo próprio navegador.

### 10.1.1 Visualização de Tópicos e Associações

Como dito anteriormente, um topic map pode ser visto como um grafo, onde os tópicos podem ser vistos como os nodos e os relacionamentos como os arcos. Cada tópico é representado como um nodo no grafo. Assim, a visualização de cada nodo – disponibilizada quando o navegador se posiciona sobre um tópico em específico – deve conter todas as suas características: seus tipos, suas instâncias, seu indicador de tema, seus nomes e suas ocorrências, além de incluir uma referência a todas as associações a qual ele faz parte.

Os nodos são rotulados usando os nomes dos tópicos. Desta forma, sempre que um nodo está relacionado com outros (sejam relações classe/instância ou associações propriamente ditas), o rótulo do nodo que está relacionado é o seu próprio nome.

Cada associação binária pode ser representada como um arco conectando os dois tópicos. Ao se posicionar em determinada associação do grafo, as informações sobre a mesma serão apresentadas. Por exemplo: em uma associação do tipo *orientar*, os tópicos participantes são visualizados (*Giovani* e *Pedro Henriques*) e os respectivos papéis de actuação (*orientando* e *orientador*), além do contexto onde ela está inserida (*ensino*). Por sua vez, os papéis de actuação são usados para designar os arcos que saem desse nodo (isto é, para descrever as associações nas quais o tópico actua).

### 10.1.2 Filtragem em Topic Maps

Um topic map pode conter milhões de tópicos e associações. Portanto é essencial seleccionar a informação relevante pois pode ser impossível mostrar todo o conjunto de tópicos e associações de uma forma eficiente.

Técnicas de filtragem de informação são necessárias para seleccionar e apresentar somente a informação relevante. A ferramenta aqui apresentada habilita os utilizadores a filtrar tópicos e associações de acordo com seu tipo ou pelos papéis de actuação em associação, por exemplo.

É possível fazer aqui uma analogia entre topic maps e mapas geográficos. Geralmente, não se encontra toda a informação desejada sobre um país em um único mapa; em vez disso, existem sub-mapas específicos de diversos contextos, tais como: mapas topográficos, mapas políticos, mapas económicos, etc.

Da mesma forma, tópicos e associações podem ser classificados em diferentes contextos. Sendo assim, sub-mapas distintos dentro de um topic map poderão fornecer as informações necessárias ao utilizador de acordo com seu interesse. Se ele estiver interessado em *teatros*, os tópicos relevantes seriam *autor*, *comédia*, *cultura*, *actor*, etc. Esta é uma forma de filtrar a informação de acordo com um contexto específico.

Um sub-mapa é o conjunto de tópicos e associações que representam um sub-domínio, dentre vários que podem existir num topic map. Um mesmo tópico pode estar inserido dentro de vários sub-mapas, de acordo com as relações que ele possui.

Uma vez filtrados os tópicos e associações, eles necessitam ser representados eficientemente. Uma sugestão para a redução do número de tipos a serem representados é a agregação de tópicos e associações com um algoritmo de classificação, como o *galois lattices* (Godin, Missaoui, and Alaoui, 1995); este algoritmo diz que se pode agrupar objectos que compartilhem propriedades comuns. Esses grupos são chamados de classes. Portanto, pode-se somente distinguir classes de tópicos na representação em vez de diferenciar todos os tópicos. Por exemplo, os tópicos *Brasil*, *América do Sul* e *Brasília* podem ser representados da mesma maneira porque eles pertencem à mesma classe *lugar*. Este mecanismo de classificação torna possível visualizar topic maps com diferentes níveis de detalhes.

Obviamente, a informação visualizada é menos precisa, mas isso pode ser aceitável dependendo do propósito da navegação. Contudo, é possível visualizar uma informação mais precisa quando o utilizador foca em uma parte específica do topic map. Por exemplo, ao estar situado no tópico *Brasília*, a informação mostrada será que *Brasília é uma cidade*, o que é mais preciso que simplesmente afirmar que *Brasília é um local*. O mesmo acontece quando verifica-se a associação entre *Brasília* e *Brasil*; a informação apresentada é *Brasília é uma cidade situada no Brasil*.

### 10.1.3 Abordagem escolhida

A fim de evitar a geração de um grafo para a visualização do topic map, o desenvolvimento do Ulisses seguiu uma abordagem paralela: a criação de um conjunto de páginas HTML,

onde cada página representa um tópico ou uma associação contido no topic map.

Os *links* determinam as relações entre os nodos; desta forma, os *links* são identificados pelos nomes dos tópicos que eles representam. Por exemplo, na página HTML gerada para o tópico *Brasília* (conforme citado na subsecção anterior), haveria a menção de uma associação com o tópico *Brasil*, onde a frase “*é uma cidade que está situada no*” representa o papel de actuação de *Brasília* perante *Brasil*.

Desta forma, a associação seria apresentada da seguinte forma:

- *Brasília* é uma cidade que está situada no *Brasil*

A palavra *Brasil*, nesta frase, possui uma ligação à página HTML que representa o tópico *Brasil*.

O algoritmo de funcionamento seguido pelo Ulisses para a criação das páginas HTML, as quais oferecerão a navegação conceptual a partir do topic map, pode ser descrito assim:

1. Definição da hierarquia de tópicos do topic map, percorrendo-se todos os tópicos e ligando-os de acordo com as relações classe/instância. Se no final desse processo verificar que não há um tópico raiz, é então criado um novo tópico que servirá de raiz para o topic map. Este novo tópico terá, como suas instâncias, todos os tópicos que não possuem tipos. Desta forma, todas as sub-árvores estarão conectadas entre si;
2. Criação de uma página inicial para o topic map, usando o nome do tópico raiz para a sua identificação e mostrando a hierarquia de tópicos no lado esquerdo da página;
3. Criação de uma página que oferece 4 formatos distintos de visualização da ontologia do topic map (tópicos abstractos), conforme será descrito na Subsecção 10.2.1;
4. Criação de uma página com o índice alfabético contendo todos os tópicos;
5. Criação de uma página para cada tópico, onde é apresentado os seus tipos, seu indicador de tema, suas ocorrências, as associações a qual ele faz parte e os tópicos que são suas instâncias. A página de cada tópico será armazenada fisicamente com nome formado por seu identificador com a extensão “html”. Deste modo, quando se faz uma referência a algum tópico, cria-se um *link* com o identificador do tópico relacionado;
6. Criação de uma página para cada associação, indicando seu tipo e seu contexto, além de seus membros e respectivos papéis de associação, onde os *links* seguem a estratégia acima: são criados a partir do identificador de cada tópico.

Seguindo esses passos, o website que corresponde a navegação conceptual do topic map será criado e armazenado numa directoria do sistema operativo.

## 10.2 Ulisses – o navegador

A ideia sobre a qual se baseou o *Ulisses* é a ideia da navegação conceptual, a qual pode ser descrita como: quando se está posicionado sobre um certo conceito, a ferramenta de navegação mostrará as informações associadas a este conceito em particular; se for escolhido algum dos outros conceitos relacionados, a navegação muda para a visão deste novo conceito; se for escolhido algum dos recursos de informação, o sistema mostrará o conteúdo do próprio recurso.

O *Ulisses* foi projectado com base em três razões principais:

- Ser um protótipo eficiente, diminuindo o tempo de criação de um navegador para *Topic Maps*;
- Criar uma ferramenta que usa *Topic Maps* que seja facilmente distribuída, instalada e usada por todos;
- Ser baseado em XML.

Em termos de implementação, foi estabelecido que a navegação seria realizada através de *web browsers*. Então, todas a informação sobre cada conceito seria apresentada em páginas Web, não necessitando assim de utilitários extras, pois todo o computador adaptado à internet teria capacidade de navegar no topic map, independente de plataforma ou sistema operativo.

Quando o topic map estiver representado em XTM, a implementação desse componente de navegação é reduzida a transformações de XML para HTML. O *Ulisses* oferece, então, duas possibilidades para o tratamento de topic maps representados em XTM. Além disso, é necessário permitir a navegação em topic maps em formato relacional (ver *BD Ontologia* na Subsecção 8.6), como os gerados pelo *Oveia*. Desta forma, o *Ulisses* possui 3 variantes, as quais estão abaixo descritas:

**Ulisses I:** transformação do XTM através de XSL (website estático) – Subsecção 10.2.1;

**Ulisses II:** transformação do XTM em tempo de execução (website dinâmico) – Subsecção 10.2.2;

**Ulisses III:** navegação em topic maps armazenados em base de dados relacionais – Subsecção 10.2.3.

Recorda-se, a propósito, o que já foi afirmado no Capítulo 6: o *Ulisses* embora tenha sido investigado no patamar filial do *Metamorphosis* para fornecer a navegação desejável sobre o topic map extraído, é um componente independente, usado além do *Metamorphosis*.

Visando tornar o *Ulisses* mais abrangente, foram adicionadas folhas de estilos XSL que permitem transformar os topic maps representados na sintaxe HyTime para a sintaxe

XTM (Ogievetsky, 2000), de modo a permitir gerar navegadores para topic maps em HyTime. Assim, a navegação conceptual pode ser gerada para os principais formatos XML para a representação de topic maps.

Além disso, o tradutor das sintaxes AsTMa= e LTM (Barta, 2002) para a sintaxe XTM pode ser utilizado para os topic maps que se encontram num destes formatos não-XML.

### 10.2.1 Ulisses I – Navegação em Topic Maps baseada em transformações XSL

A variante mais utilizada do Ulisses é aquela que gera o website de uma só vez, isto é, transforma previamente (antes de se fazer a navegação) o topic map, em XTM, num conjunto de páginas HTML que não voltam a ser geradas em toda a navegação (site estático) (Librelotto, Ramalho, and Henriques, 2003c). Por isso, esta variante do Ulisses é desenvolvida em XSL (Thompson, 2003), criando websites em HTML com algumas componentes em *Javascript*.

O algoritmo de funcionamento desta variante do Ulisses foi descrito na Secção 10.1.3.

Esta versão do Ulisses oferece opções que permitem rápidas modificações em todo o website, como por exemplo, alterações em termos de *layout* (cores e tamanho das fontes), comentários a serem incluídos nas páginas criadas e a inserção de imagens.

A seguir, cada formato de página gerado pelo Ulisses é apresentado, nomeadamente: a página inicial e a página que mostra a ontologia do topic map, a página que descreve um tópico e, por fim, o modelo de página para as associações.

#### Página principal do topic map

A página inicial (*home page*) da navegação num topic map no Ulisses é apresentada na Figura 10.1. Esta imagem expõe no seu lado esquerdo, a hierarquia de classes obtida a partir da ontologia, na forma de um menu que permite aceder qualquer tipo de tópico do topic map – item (1).

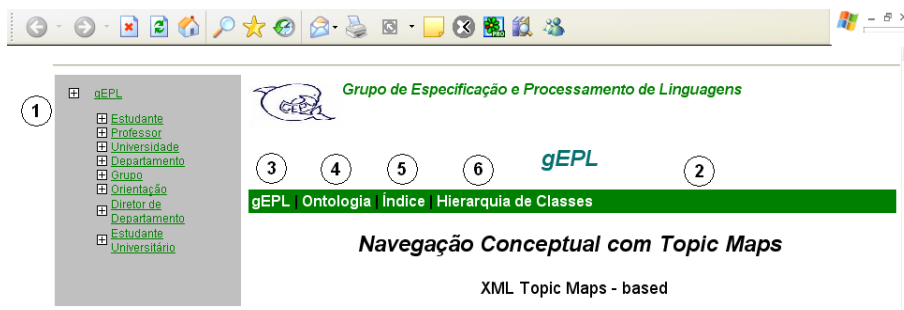


Figura 10.1: Visualização inicial de um topic map no *Ulisses*

No corpo do navegador, há uma barra de navegação – item (2) – que se encontra em todas

as páginas geradas pelo Ulisses. Esta barra oferece 4 opções navegacionais:

**Home:** a primeira opção faz com que a navegação volte à página inicial do topic map, independente em que nodo a visualização do topic map se encontre num determinado instante. Essa opção é referenciada pelo nome do tópico raiz do topic map. Na Figura 10.1, *gEPL*<sup>1</sup> é o nome do tópico raiz, por isso aparece na barra de navegação em todas as páginas geradas para este topic map – item (3);

**Ontologia:** ligação que remete à página que descreve a hierarquia de classes da ontologia do topic map – item (4);

**Índice:** indica a página que apresenta um índice alfabético dos tópicos – item (5);

**Hierarquia de Classes:** propicia a visualização da hierarquia de classes contida na ontologia do topic map em um formato gráfico – item (6).

### Menu Ontologia

A ontologia – item (3) na Figura 10.2 – pode ser descrita em 4 partes, de acordo com as opções do sub-menu mostrado na Figura 10.2 e descritos abaixo:

**Temas:** apresenta a árvore hierárquica completa formada pelos tipos de tópicos e suas instâncias – item (1);

**Relacionamentos:** mostra todas as associações encontradas no topic map – item (2);

**Papeis de Actuação:** descreve os papeis de actuação e as associações nas quais cada papel está envolvido – item (3);

**Recursos de Informação:** lista todas as ocorrências encontradas no topic map agrupadas por seus tipos – item (4).

### Visualização de um tópico

A visualização de um tópico pode ser composta por 6 partes, conforme pode ser visto na Figura 10.3, onde se apresenta a visualização do tópico *Giovani R Librelotto*:

**Nome do topic map e barra de navegação:** no cabeçalho do topic map aparece o nome do tópico raiz (neste caso, *Giovani R Librelotto*). Abaixo, encontra-se a barra de navegação, com as opções de voltar à página principal, aceder a ontologia, o índice geral e a hierarquia de classes do topic map – item (1);

---

<sup>1</sup>*gEPL* refere-se ao Grupo de Especificação e Processamento de Linguagens da Universidade do Minho. Ver: <http://wiki.di.uminho.pt/wiki/bin/view/EPL/WebHome>

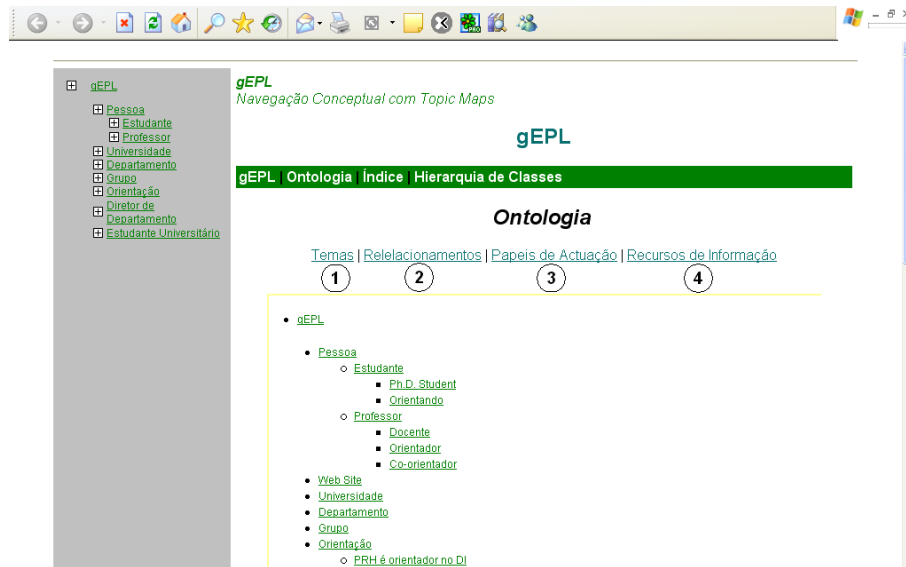


Figura 10.2: Visualização da ontologia de um topic map no *Ulisses*

**Tipo do tópico:** é uma ligação ao tipo do tópico em questão. No caso, o tópico *Giovani R Librelotto* é uma instância de *Estudante* – item (2);

**Nome do Tópico:** em destaque, o nome do tópico caracteriza o tópico em questão – item (3);

**Recursos de Informação:** as ocorrências do tópico, juntamente com seus tipos, são apresentadas nesta parte. Por exemplo: uma URL é definida como uma referência a um recurso do tipo *website* que caracteriza o tópico *Giovani R Librelotto* – item (4);

**Tópicos Relacionados:** os tópicos que estão associados, assim como o papel de actuação do tópico em questão em cada associação, se encontram postos de forma que a leitura seja intuitiva. Por exemplo: *Giovani R Librelotto é orientado por Pedro R Henriques* define uma associação envolvendo dois tópicos (*Giovani R Librelotto* e *Pedro R Henriques*), além do papel de actuação do primeiro em relação ao segundo (*é orientado por*) – item (5);

**Instâncias:** lista dos tópicos que são instâncias do tópico em questão. Por exemplo: a visão do tópico *Estudante* apresenta o tópico *Giovani R Librelotto* e suas demais instâncias. Na Figura 10.3 não se encontra nenhuma instância, pois o tópico *Giovani R Librelotto* é um nodo folha na hierarquia de classes do topic map.

Dessa forma, todas as características de tópicos discutidas na Subsecção 10.1.1 encontram-se descritas numa página que segue o formato acima descrito.



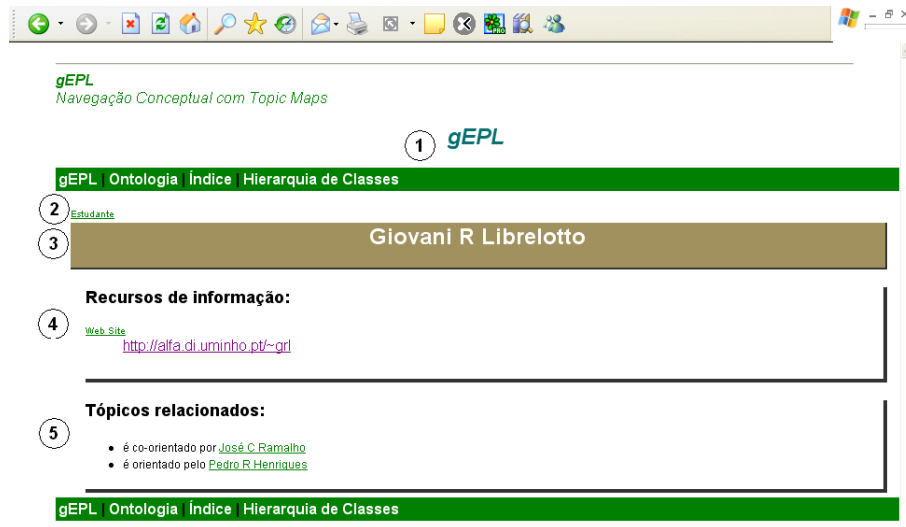


Figura 10.3: Visualização de um tópico no *Ulisses*

### Visualização de uma associação

A visualização de uma associação pode ser composta por 5 partes, conforme pode ser visto na Figura 10.4, onde se apresenta a visualização da associação *Chefe de Grupo*:

**Nome do topic map e barra de navegação:** assim como acontece na visualização de tópicos, o nome do tópico aparece no alto da página (neste caso, *Chefe de Grupo*); logo abaixo encontra-se a barra de navegação – item (1);

**Tipo da associação:** é uma ligação ao tipo da associação. No caso, a associação *Chefe de Grupo* é uma instância de *gEPL*, o qual é o tópico raiz do topic map. Esse facto indica que a associação não possuía nenhum tipo e foi relacionado, pelo *Ulisses*, com o tópico raiz – item (2);

**Nome da Associação:** em destaque, o nome que caracteriza a associação em questão – item (3);

**Tipos dos Tópicos Membros:** em cada frase que determina a relação entre os tópicos membros, os seus tipos são citados; cada citação é um *link* para a página referente ao tipo de tópico – item (4);

**Membros:** os membros da associação constituem *links* para a páginas dos referidos tópicos. Por exemplo, o tópico *José C Ramalho* participa da associação *Chefe de Grupo* juntamente com o tópico *Grupo de Especificação e Processamento de Linguagens* – item (5);

**Papeis de Actuação:** cada membro da associação está relacionado aos demais membros através de um papel de actuação. Por exemplo, o tópico *José C Ramalho* possui

o papel é *chefe de* na associação com o tópico o tópico *Grupo de Especificação e Processamento de Linguagens* – item (6).



Figura 10.4: Visualização de uma associação no *Ulisses*

Assim como ocorre com os tópicos, as associações também tem todas as suas características apresentadas em uma página que segue o formato acima descrito.

### 10.2.2 Ulisses II – Navegação em Topic Maps com transformação em tempo de execução

As transformações em tempo de execução são implementadas com transformações XML, auxiliadas por uma CGI (*Common Gateway Interface*). A página inicial é uma chamada a uma CGI parametrizada à raiz do topic map (que vai servir de ponto de entrada para a navegação). A CGI aplica a transformação no topic map e produz uma visão HTML. Nessa visão, todos os *links* gerados são chamadas à mesma CGI mas com diferentes parâmetros, que indicam a localização da nova posição, ou seja, permitirá a visão do tópico relacionado.

Esta é a solução mais prática porque somente necessita de dois ficheiros: o topic map (especificado de acordo com a sintaxe XTM) e a CGI. Contudo, as transformações em tempo de execução consomem muito tempo, pois o topic map é interpretado em tempo de execução. Desta forma, a cada nova chamada a CGI, o topic map deve ser interpretado mais uma vez, para formar a visualização a ser disponibilizada no navegador. Por isto, esta solução tem sido utilizada apenas com topic maps considerados pequenos, ou seja, com um número inferior a 200 tópicos e associações.

A navegação no Ulisses baseado em CGI segue o modo descrito na Subsecção 10.2.1. Apesar da navegação ser gerada de uma forma distinta – em vez de uma página HTML para cada tópico e associação, usa-se uma chamada à CGI com o identificador do tópico ou associação desejado – a visualização do topic map é a mesma. Assim, as imagens apresentadas na subsecção anterior servem também para descrever essa variante do Ulisses.

Assim como a variante do *Ulisses* apresentada na subsecção anterior, esta variante baseada em CGI também permite a navegação sobre topic maps representados de acordo com a sintaxe XTM. Ou seja, se o topic map estiver em algum outro formato, deve ser convertido para XTM (antes da geração da navegação) pelos processadores de HyTime (Ogievetsky, 2000), AsTMa= e LTM (Barta, 2002).

### 10.2.3 *Ulisses* III – Navegação em Topic Maps armazenados na BD Ontologia

Conforme apresentado na Secção 8.6, além da geração de topic maps em ficheiros XTM, o *Oveia* propicia também o armazenamento dos mesmos num formato relacional. Desta forma, em vez de ficheiros texto, uma base de dados, chamada *BD Ontologia*, funciona como um repositório de topic maps. A partir dessa base de dados, é possível navegar nos topic maps utilizando consultas SQL.

As consultas SQL podem ser realizadas para diferentes fins. Um exemplo pode ser visto no código abaixo, o qual apresenta um comando SQL para retornar as instâncias de um tópico, que se encontra na base de dados.

```
1 | SELECT bn.baseNameString, t.* FROM topicRef tr
2 |     inner join instanceOf [io] on tr.topicRefID = [io].topicRefID
3 |     inner join instanceOfTopic iot on [io].instanceOfID = iot.instanceOfID
4 |     inner join topic t on t.topicID = iot.topicID
5 |     inner join baseName bn on t.topicID = bn.topicID
6 | WHERE tr.topicID = 1 AND bn.scopeID is null
7 | ORDER BY bn.baseNameString
```

A consulta SQL acima busca, na *BD Ontologia*, todos os tópicos que são instâncias de um determinado tópico, ordenando-os pelo nome. Assim, entende-se que há uma consulta SQL implementada para cada tipo de interrogação que se pode vir a fazer no topic map, seja ela sobre tópicos, associações ou ocorrências.

As principais vantagens deste formato de navegação são:

- manter o topic map armazenado num modelo relacional. Pode-se, assim, aproveitar os recursos que uma base de dados relacional oferece para a gestão do topic map;
- realizar a navegação sobre o topic map actual; isso significa que se houver mudanças no topic map armazenado, elas automaticamente serão reflectidas na visualização;
- alternar o topic map a ser visualizado, sem necessitar abrir um novo navegador. Como uma única *BD Ontologia* pode armazenar diversos topic maps, é possível alternar o topic map em que se está a navegar, de uma maneira simples.

Contudo, essa variante do *Ulisses* também possui algumas desvantagens, tais como:

- necessita de um gestor de bases de dados (onde está o topic map a ser navegado) e de um servidor Web (por exemplo, o TomCat (Apache, 2003)) para o seu completo funcionamento;

- a velocidade da navegação depende de factores externos, tais como a base de dados e o servidor Web, pois as páginas são montadas em tempo de navegação.

### 10.3 Trabalhos Relacionados com o Ulisses

Actualmente, os navegadores (*browsers*) para Topic Maps mais conhecidos pela comunidade académica de Semantic Web são: Omnigator (Ontopia, 2002b), Topic Map Designer (Heckel, 2001a), xSiteable (Johannesen, 2004) e TMNav (SourceForge, 2005).

Actualmente, o Ontopia Omnigator talvez seja o navegador de topic maps mais difundido. O Omnigator é uma aplicação que permite carregar e navegar sobre qualquer topic map, usando um browser para a Web. O objectivo do desenvolvimento do Omnigator foi incentivar o uso de Topic Maps, ensinando os princípios básicos do paradigma.

O Omnigator faz parte do Ontopia Knowledge Suite (OKS) (Ontopia, 2004). Isto implica que algumas funcionalidades do Omnigator foram projectadas para trabalhar em conjunto com outras ferramentas, tais como o Ontopia Navigator Framework (Ontopia, 2002a). Contudo, o Omnigator é o único módulo integrante do OKS fornecido livremente<sup>2</sup>; os demais módulos devem ser adquiridos. Outro inconveniente do Omnigator é que a versão disponibilizada livremente está limitada a 5000 tópicos e associações; desta forma, quando um topic map possuir uma quantidade superior a este limite, a navegação não é permitida ao utilizador.

Apesar deste inconveniente comercial e da limitação imposta na versão distribuída livremente, o Omnigator é considerado uma ferramenta completa e eficiente, que produz um navegador com uma interface muito agradável e simples de manusear, como se percebe na Figura 10.5. Contudo, o Omnigator é um interpretador – processa o topic map quando carrega a especificação XTM indicada – o que requer a sua presença na máquina do utilizador final, ou então o recurso a um servidor Web que terá de estar activo numa determinada máquina.

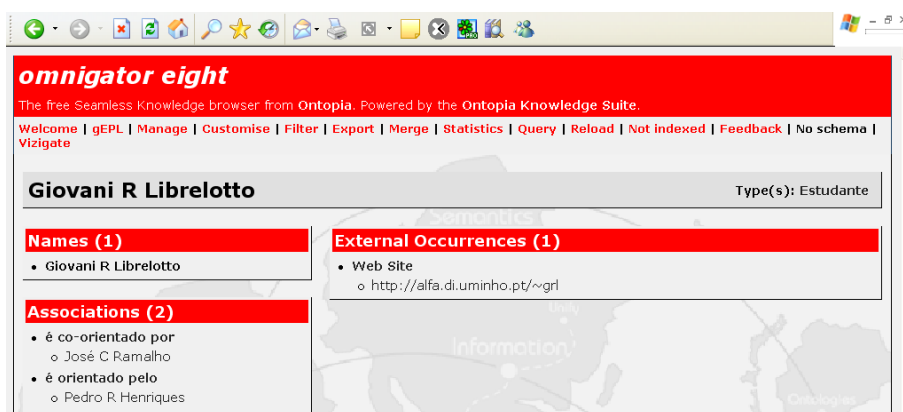


Figura 10.5: Visualização de um tópico no Omnigator

<sup>2</sup>Disponível em: <http://www.ontopia.net/omnigator/models/index.jsp>

Constata-se, na prática, que o *Omnigator* é uma óptima solução para a fase de desenvolvimento, quando o topic map está constantemente a ser alterado<sup>3</sup>. Nessa fase, a navegação proporcionada pelo *Omnigator* pode ser utilizada para certas verificações, como por exemplo, a correcção dos nomes dos tópicos e dos seus contextos. Porém, numa situação de produção, em que o topic map está estável, é preferível usar uma solução em que o visualizador possa ser aberto por qualquer *browser* sem requerer a presença do processador.

Foi precisamente este requisito que nos levou a desenvolver o *Ulisses* que, sendo comparável a um compilador, gera um conjunto de páginas HTML estáticas. Assim, o *Ulisses* é invocado uma vez (pela equipa responsável pelo desenvolvimento do navegador) e o resultado permite efectuar navegações com qualquer *browser* que o utilizador final escolha.

Entretanto, uma vantagem significativa do *Omnigator* é o seu módulo chamado *Vizigator* (Gennusa, 2004). O *Vizigator* propicia uma navegação gráfica de topic maps fornecendo:

- uma visão geral da estrutura da informação, de forma intuitiva e instantânea;
- uma travessia no grafo formado pelo topic map;
- a habilidade de ver e entender os relacionamentos;
- visões em diferentes níveis de granularidade;
- acesso intuitivo a modelos de dados não-familiares ao utilizador.

Para exemplificar, a Figura 10.6 apresenta a visualização de um topic map no *Vizigator*. Nesta figura percebe-se o grafo formado pelo conjunto de tópicos e associações, e a partir dele, verificar quais são os tópicos que estão relacionados com o tópico *Giovani R Librelotto*. Nesse caso, observa-se que há uma associação entre ele e cada um dos tópicos *Pedro R Henriques* e *José C Ramalho*; além disso, o tópico *Estudante* define o seu tipo.

A visualização gráfica fornecida pelo *Vizigator* se caracteriza, portanto, num diferencial a favor do *Omnigator*, por propiciar duas navegações distintas sobre *Topic Maps*: numa visão HTML contendo toda a informação sobre cada tópico (como faz o *Ulisses*) ou numa visão gráfica, apresentando o grafo propriamente dito (como faz o *Topic Map Designer*).

O *Topic Map Designer* (Heckel, 2001b) é um aplicativo que permite a edição de *Topic Maps* e sua consequente visualização gráfica. Contudo, ele não tem como objectivo ser um ambiente completo para a criação de *Topic Maps* e suporta somente topic maps no formato HyTime. De um modo geral, é aconselhado seu uso para topic maps pequenos (não maiores de 50 tópicos).

O *xSiteable* (Johannesen, 2004) é uma ferramenta de desenvolvimento de websites criada em XSLT, com um pacote de administração PHP. De modo geral, possui características muito similares ao *Ulisses*, no que diz respeito à formação do website, representação das

---

<sup>3</sup>Como o *Omnigator* interpreta a especificação na hora em que o serviço é requisitado, qualquer alteração que tenha sido produzida no ficheiro XTM é logo reflectida no visualizador.

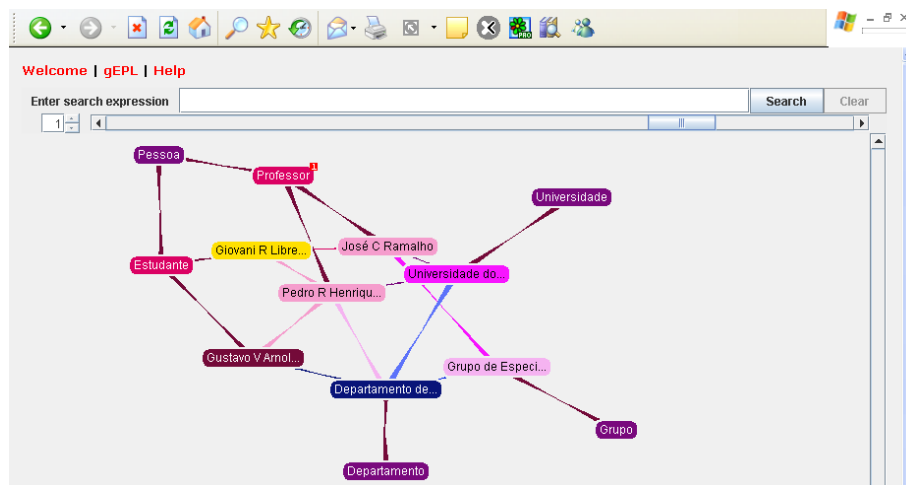


Figura 10.6: Visualização de um tópico no Omnigator

características dos tópicos e associações, assim como na visualização de todas estas informações em uma página HTML. Ambas utilizam tecnologia XSLT para a geração de páginas HTML, a partir de um documento XTM.

Porém, o xSiteable apenas possui a opção de criar uma página HTML por tópico. Ao contrário do *Ulisses*, o xSiteable não permite a criação de um website dinamicamente, (conforme apresentado na Subsecção 10.2.2). Além disso, não suporta qualquer outro formato de topic maps, além de XTM e CSXTM<sup>4</sup> (Johannesen, 2003). O *Ulisses*, por sua vez, permite a navegação em topic maps armazenados na BD Ontologia.

A mesma vantagem acima – permitir a navegação de topic maps armazenados num modelo relacional – o *Ulisses* possui sobre o TMNav. O TMNav (SourceForge, 2005) é uma aplicação *Java/Swing* para propiciar a navegação em topic maps que utiliza uma interface baseada em grafos. Ele funciona sobre o aplicativo TM4J (SourceForge, 2003b), permitindo assim uma navegação em páginas Web ou num grafo dinâmico que utiliza a biblioteca Touch-Graph (Shapiro, 2002). Com isso, a desvantagem do *Ulisses* é justamente o facto de não fornecer uma visualização diagramática do grafo, como faz tanto o TMNav, como o Omnigator/Vizigator.

## 10.4 Sumário sobre o Ulisses

Em resumo, o *Ulisses* representa uma ferramenta com objectivos claros (navegação conceptual em Topic Maps), baseada em conceitos de navegação em grafos, a qual utiliza tecnologias bastante difundidas para sua implementação, não o tornando dependente de plataforma ou aplicativo específico.

<sup>4</sup>CSXTM é um formato de representação de Topic Maps compacto, simplificado e baseado em XTM. Desenvolvido pelos mesmos criadores do xSiteable, utiliza-se folhas de estilos XSL para a conversão de XTM para CSXTM, e vice-versa.

Actualmente, o **Metamorphosis** tem sido utilizado em vários projectos de pequena e média dimensão. Até ao momento, provou ser uma boa ferramenta de prototipagem. Com isso, a necessidade de uma ferramenta que disponibilize uma navegação conceptual é importante e indispensável, pelo que se acrescentou o **Ulisses** ao **Metamorphosis**. Assim, as interfaces Web – que disponibilizam o conhecimento representado no topic map – são criadas rapidamente e sem grandes dificuldades por parte dos utilizadores.

O **Ulisses** fornece uma navegação completa sobre topic maps. A representação do conhecimento é apresentado de uma forma simples e precisa, formando uma rede semântica baseada em tópicos e associações. Quando se acede às informações referente a um determinado tópico, visualizam-se suas características (o seu tipo, suas instâncias, seus identificadores de tema, seus nomes e suas ocorrências) e as associações relacionadas com este tópico (incluindo os papéis de associação actuado por ele e os tópicos associados).

Apresentando três tipos distintos de navegadores, o **Ulisses** abrange praticamente todos os formatos de representação de **Topic Maps**. Cada navegador possui sua aplicação ideal: o navegador baseado em XSL é indicado para qualquer tamanho de topic maps, pois gera uma interface completa, independente da quantidade de tópicos e associações (**Ulisses I**); o navegador baseado em CGI é indicado para uma rápida visualização de pequenos topic maps (com até 200 tópicos e associações) (**Ulisses II**); por fim, o navegador para base de dados relacional acrescenta a possibilidade de navegação em topic maps armazenados na BD Ontologia (**Ulisses III**).

A fim de deixar a critério do utilizador escolher qual é o formato de navegação que mais lhe interessa, o **Oveia** (Capítulo 8) permite a extracção de um topic map armazenado na BD Ontologia para o formato XTM, pois esta sintaxe é definida na norma ISO 13250 como sendo o formato ideal para o intercâmbio de topic maps entre aplicações. Assim, um topic map na BD Ontologia pode facilmente ser convertido para um formato normalizado, podendo assim ser processado por outros motores de **Topic Maps**.





## Parte III

# Topic Maps: Casos de Estudo



## Capítulo 11

# O Ulisses aplicado à navegação em Mapas de Conceitos

*Os Pássaros*

-

*Olho em silêncio os pássaros que passam,*

*A luz é veloz nos seus olhos de ave*

*Os planos oblíquos*

-

*Nada lembro já*

*Não espero ainda:*

*Olho em silêncio os pássaros que passam*

*Escolho o inverno*

Luis Soares Barbosa

O presente capítulo pretende apresentar as potencialidades do *Ulisses*, em conjunto com *Topic Maps*, como uma parceria que providencia oportunidades de aprendizagem. Para isto, defende-se o uso da norma *Topic Maps* para a descrição de Mapa de Conceitos (MC), como forma de organizar o conhecimento que constitui o conteúdo de uma disciplina de *Introdução à Informática para Ciências Sociais (IIPCS)*, e a utilização do *Ulisses* para proporcionar a navegação conceptual desejada.

Este capítulo então mostra que Mapas de Conceitos e *Topic Maps* possuem a mesma finalidade – *representar o conhecimento* – mas originados por correntes de investigação diferentes <sup>1</sup>.

Sugere-se então a utilização da notação XTM (XML Topic Maps) para representar MC e o *Ulisses* para criar uma interface Web que permita, simultaneamente, visitar os conceitos percorrendo as ligações que os unem e aceder aos materiais didácticos associados a alguns desses conceitos e também disponíveis na Web.

---

<sup>1</sup>Por um lado temos os agentes educativos, por outro os tecnólogos.

A argumentação apresentada a favor desta escolha tecnológica assenta na utilização do Ulisses para a derivação automática de um navegador sobre os conceitos encontrados na especificação XTM referente ao Mapa de Conceitos. Tal navegação permite, por um lado, ir de conceito em conceito analisando as várias relações que os ligam e, por outro lado, possibilita o acesso à informação final associada aos conceitos básicos.

Para atingir esses objectivos, este capítulo está estruturado da seguinte maneira: na Secção 11.1 introduzem-se os Mapas de Conceitos, do ponto de vista dos agentes educativos; a ponte entre Mapas de Conceitos e Topic Maps será feita na Secção 11.2; o caso de estudo propriamente dito será apresentado na Secção 11.3; a aplicação concreta do Ulisses ao caso de estudo será descrito na Secção 11.4, onde será demonstrado a variante I do Ulisses; finalmente, na Secção 11.5, será feita uma breve reflexão sobre os benefícios que se retiram de projectos multi-disciplinares como este que se descreve ao longo do capítulo e comentam-se os próximos passos para lhe dar o prosseguimento desejado.

## 11.1 Mapas de Conceitos

Os Mapas de Conceitos (MC) têm a sua origem no movimento da teoria construtivista da aprendizagem, de Ausubel (Ausubel, 1968). A ideia central desta teoria é da aprendizagem significativa, que surge em oposição à de aprendizagem mecânica, em que o conhecimento é memorizado sem que o aluno estabeleça relações entre a nova informação e aquela que já existe na sua estrutura cognitiva. Assim, a aprendizagem significativa é o processo através do qual uma nova informação se relaciona com um aspecto relevante da estrutura do conhecimento do sujeito (Ausubel, 1968). O aluno ao confrontar-se com novos conhecimentos deve ser capaz de distinguir os principais conceitos e de os relacionar com os conhecimentos anteriores.

Os Mapas de Conceitos foram usados pela primeira vez (há mais de 4 décadas) por Novak, o qual defendia que o mapa de conceitos pode ter várias funções em simultâneo: como recurso de auto-aprendizagem ao dispor dos alunos e professores; como método para encontrar e explicitar significado para os materiais de estudo; e como estratégia que estimula a organização dos materiais de estudo. Os MC são, para Joseph Novak (Novak, 1977), uma ferramenta para organizar e representar conhecimento.

Hoje em dia, os mapas de conceitos são utilizados com variados fins pedagógicos:

- como uma técnica de planificação;
- como forma de estruturar conteúdos para a leccionação de determinada unidade didáctica (que será o exemplo apresentado ao longo deste capítulo);
- para sintetizar informação;
- para consolidar informação a partir de diferentes fontes de pesquisa;
- como meio de simplificar a abordagem a problemas complexos;

- para ajudar o aluno a fixar a sua atenção nos conceitos mais importantes, ajudando-o também a aprender a representar ideias de uma maneira gráfica;
- como suporte à avaliação formativa;
- como instrumento de apoio durante a exposição de um conteúdo;
- como meio diagramático de esclarecer (ou descrever) ideias que as pessoas têm sobre um determinado assunto;
- etc.

Um MC vai muito além de um esquema convencional desenhado ad-hoc, informalmente; ele tem uma semântica precisa que pode ser descrita formalmente por um grafo que é constituído por nós, onde se inscrevem os conceitos (substantivos no singular), e ramos (ligações), que representam as relações (verbos transitivos) entre conceitos<sup>2</sup>. Assim, o MC pode ser lido caminhado no grafo e formando proposições numa linguagem explícita e concisa. De igual modo, também as frases da nossa linguagem comum podem facilmente ser traduzidas para o MC; por exemplo, na frase *squash é um desporto*, “*squash*” e “*desporto*” são dois conceitos ligados através de um relacionamento “*é um*”. O mapa de conceitos que representa esta frase é o mais simples possível, sendo constituído por dois nós conectados por uma ligação, conforme se pode ver na Figura 11.1. Estes relacionamentos são nominativos, ou seja, cada relacionamento entre dois conceitos forma uma proposição.

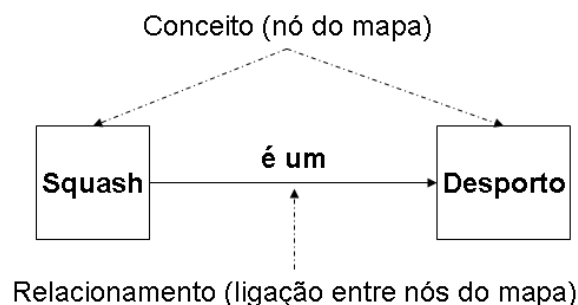


Figura 11.1: Exemplo de um Mapa de Conceitos simples

Já a Figura 11.2 ilustra um caso bem mais complexo; trata-se de um mapa de conceitos concreto que pretende descrever o programa lectivo de IIPCS. O termo *Programa Lectivo de IIPCS* é o nó de topo (ou inicial) do mapa apresentado, sendo composto por 3 Unidades, onde cada unidade corresponde a um módulo da disciplina: *Processamento de Documentos*, *Processamento Numérico* e *Processamento de Informação*.

Ao construir um MC há total liberdade para a escolha do tipo de relações que se quer mostrar a ligar os conceitos. Contudo uma das relações sempre presente, e que é fundamental, é a relação *é um* que permite criar uma hierarquia, com conceitos mais gerais e inclusivos no topo do mapa e os mais específicos, portanto, os conceitos menos gerais,

<sup>2</sup>Estas ligações cruzadas ajudam a se perceber como alguns domínios de conhecimento se relacionam entre si.

dispostos hierarquicamente por baixo. Esta relação cria a noção de *inclusão de classes*, ou seja de *subclasse* (à frente designada por *instância*) e *classe*.

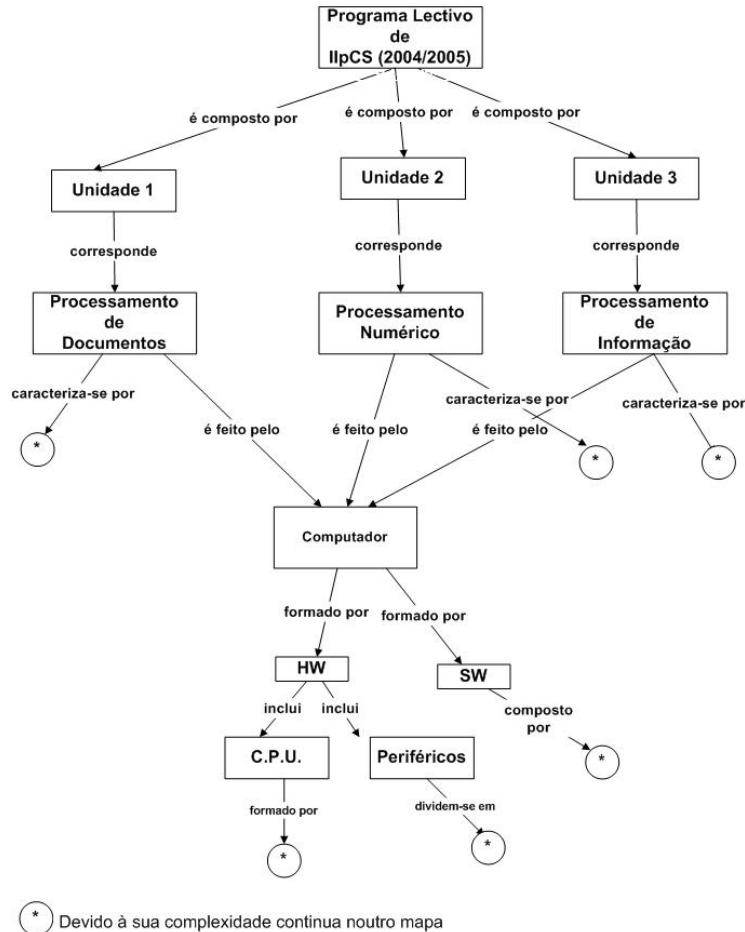


Figura 11.2: Um Mapa de Conceitos referente à disciplina de IIPCS

Em resumo, MC são instrumentos capitais para o professor programar uma disciplina, para recolher os materiais didácticos de apoio e para preparar cada aula. São também importantes para o aluno entender os objectivos da aula e aferir os conhecimentos adquiridos; segundo vários autores, a avaliação é precisamente um processo de comparação entre mapas de conceitos: o proposto pelo professor e o construído pelo aluno.

## 11.2 Mapas de Conceitos em Topic Maps

Os Mapas de Conceitos (MC) têm sido usados em inúmeras áreas científicas, da educação à política, da filosofia às ciências ditas exactas, para construir representações visuais de estruturas de conhecimento, de fenómenos científicos complexos e até de ideias em fase embrionária.

No seguimento de experiências efectuadas, concluiu-se (Pacheco et al., 1999) que a organização de conteúdos curriculares com apoio em MC é uma das abordagens que tem, nos dias de hoje, mais sucesso para esquematizar o tipo de ensino que se pretende. Como se dirá à frente, os MC têm outras importantes vantagens tanto para o professor como para o aluno. Apesar de tudo isso, constata-se que o seu uso e exploração está muito aquém do que seria de esperar e desejável.

Isso deve-se, principalmente, ao esforço requerido para desenhar mapas reais no papel, que tendem a ser grandes e complexos, e à inerente dificuldade em navegar posteriormente nesse mapa acedendo aos materiais didácticos associados aos conceitos. Recentemente assistiu-se ao aparecimento de vários livros em que surgem MC no início de cada capítulo, como forma de sintetizar a matéria aí incluída, mas como é fácil de perceber, não há uma interligação directa entre os nós do mapa e os locais do livro onde os conceitos são explicados.

Vivendo-se numa era em que existe uma preocupação constante na integração das novas tecnologias no processo ensino/aprendizagem, a tecnologia Topic Maps vem permitir que os mapas de conceitos tradicionais saltem do papel para o écran do computador, trazendo as tais capacidades de exploração que no mundo físico eram desejadas mas impossíveis de criar!

A semelhança entre um mapa de conceitos e uma ontologia e o facto de os Topic Maps estarem, actualmente, a ser usados para exprimir ontologias, levou-nos a pensar em *usar TM para mostrar e explorar MC na Internet*.

Para fazer a transposição de Mapas de Conceitos (apresentados na Secção 11.1) para Topic Maps, usando a sintaxe XTM, procedemos de modo sistemático como se vai explicar através da apresentação de um Caso de Estudo (CE): *Topic Map para representar o Mapa de Conceitos de uma disciplina de Introdução à Informática*.

O CE escolhido está associado a um projecto, no contexto concreto da disciplina de Introdução à Informática de uma licenciatura em Ciências Sociais, que nasceu da necessidade de criar um sítio WWW moderno para apoio eficiente e eficaz ao ensino presencial tradicional. Pretendia-se usar a clássica “*página WWW da disciplina*” (elaborada para satisfazer os requisitos oficiais<sup>3</sup> do “Dossier de Disciplina”) de modo a construir esse suplemento educativo com suporte na Web.

O novo sítio seguirá integralmente, em termos de interface e funcionalidade, o actualmente existente<sup>4</sup>, apenas diferindo do actual pelo facto de vir a ser gerado dinamicamente a partir de uma base de dados com toda a informação sobre a disciplina. Nesse âmbito, decidiu-se integrar no sítio um mapa de conceitos descrevendo o conteúdo curricular da disciplina. Para isso necessitávamos de um visualizador que permitisse navegar sobre o mapa e, a partir dele, aceder aos recursos de informação disponibilizados para suportar o dito conteúdo curricular.

---

<sup>3</sup>Há vários anos em vigor na Universidade do Minho.

<sup>4</sup>Acessível em [www.di.uminho.pt/~prh/IIA04/](http://www.di.uminho.pt/~prh/IIA04/).

### 11.3 Um Topic Map para representar um Mapa de Conceitos

Para criar o *topic map* para representar o *Mapa de Conceitos* de uma disciplina de *Introdução à Informática para Ciências Sociais*, o primeiro passo consistiu na análise paralela do mapa de conceitos fornecido<sup>5</sup> e do esquema da sintaxe XTM a usar.

Depois, procedeu-se metodicamente a um processo de produção de *Topic Maps*, formado por 4 etapas<sup>6</sup>, que se discute nas próximas subsecções.

#### 11.3.1 Escolha da raiz (ou símbolo inicial) do Topic Map

Para representar o símbolo inicial do topic map referente a este caso de estudo, é criado um tópico com o nome *Introdução à Informática para as Ciências Sociais*, identificado por IIPCS. Este tópico é então escolhido pelo Ulisses como a raiz da navegação, pois seu indicador de tema (linha 6) faz referência ao identificador do topic map (id="first" – linha 2), determinando assim que este tópico define o título da página principal a ser gerada. A especificação do cabeçalho do topic map, contendo seus *namespaces* (linhas 2 e 3) e o tópico identificado por IIPCS (linhas 4 a 11) está apresentado a seguir:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <topicMap id="first" xmlns="http://www.topicmaps.org/xtm/1.0/"
3   xmlns:xlink="http://www.w3.org/1999/xlink">
4   <topic id="IIPCS">
5     <subjectIdentity>
6       <subjectIndicatorRef xlink:href="#first"/>
7     </subjectIdentity>
8     <baseName>
9       <baseNameString>Introdução à Informática para Ciências Sociais</baseNameString>
10    </baseName>
11  </topic>
12  ...
13 </topicMap>

```

As reticências (“...” – linha 12) indicam que outros tópicos e associações serão acrescentados a este topic map, os quais são definidos nas próximas subsecções.

#### 11.3.2 Definição dos tópicos correspondentes aos conceitos do MC

A segunda fase compõe-se da definição dos tópicos. Para isso percorre-se o MC a fim de identificar todos os conceitos existentes e analisar as suas relações classe/subclasse. Conforme visto no Capítulo 3, a sintaxe XTM permite a definição deste tipo de relação no elemento *instanceOf*, onde o seu conteúdo é o identificador do tipo deste tópico.

Para exemplificar, os 2 tópicos abaixo possuem uma relação classe/subclasse, pois o tópico *Unidade Lectiva* (identificado por *unidadeLectiva*) é definido como tipo do tópico *Proces-*

<sup>5</sup>Na forma de um grafo desenhado em Visio®, ocupando 5 páginas A4 cheias.

<sup>6</sup>A ordem entre a 3ª (Subsecção 11.3.3) e a 4ª (Subsecção 11.3.4) fases é arbitrária.



*samento de Documentos* (identificado por `procDocumentos`), de acordo com a referência apresentada na linha 8.

```

1 | <topic id="unidadeLectiva">
2 |   <baseName>
3 |     <baseNameString>Unidade Lectiva</baseNameString>
4 |   </baseName>
5 | </topic>
6 | <topic id="procDocumentos">
7 |   <instanceOf>
8 |     <topicRef xlink:href="#unidadeLectiva"/>
9 |   </instanceOf>
10 |   <baseName>
11 |     <baseNameString>Processamento de Documentos</baseNameString>
12 |   </baseName>
13 | </topic>

```

No final deste passo, a hierarquia de classes do topic map estará construída. O passo seguinte define as associações existentes entre os conceitos transformando o topic map, até então representado em uma árvore formada pelas relações classe/subclasse, em um grafo.

### 11.3.3 Definição das associações correspondentes às relações do MC

A partir do momento em que todos os conceitos estiverem representados por tópicos, passa-se à 3ª fase: a definição das associações. Esta fase é realizada em 4 passos:

1. determinar quais tópicos possuem relacionamento entre si;
2. definir qual o papel de actuação de um tópico em relação aos demais em cada associação;
3. criar um tópico que representa o tipo de cada associação, contendo os papéis de actuação e sua descrição textual (para ser visualizada na navegação);
4. declarar cada associação, definindo seu tipo, seus membros e respectivos papéis de actuação.

Como exemplo, toma-se dois tópicos – `computador` e `documento` – que possuem uma relação entre si, do tipo *processar*. Para definir essa associação, identifica-se inicialmente os papéis de actuação de cada um dos membros.

Os papéis também são vistos como tópicos; portanto, terão também de ser mapeados para a sintaxe XTM. Desta forma, o código XTM abaixo apresenta os tópicos membros que fazem parte da associação *processar* – *documento* e *computador* (linhas 1 a 10) – e os seus respectivos papéis de actuação – *processado* e *processa* (linhas 11 a 20).

```

1 | <topic id="documento">
2 |   <baseName>
3 |     <baseNameString>Documento</baseNameString>
4 |   </baseName>

```

```

5 |</topic>
6 |<topic id="computador">
7 |  <baseName>
8 |    <baseNameString>Computador</baseNameString>
9 |  </baseName>
10|</topic>
11|<topic id="processado">
12|  <baseName>
13|    <baseNameString>é processado por</baseNameString>
14|  </baseName>
15|</topic>
16|<topic id="processa">
17|  <baseName>
18|    <baseNameString>processa</baseNameString>
19|  </baseName>
20|</topic>

```

O tópico que caracteriza o tipo da associação *processar*, designado por **Associação entre documento e computador**, está abaixo definido.

```

1 |<topic id="assoc_doc_comp">
2 |  <baseName>
3 |    <baseNameString>Associação entre documento e computador</baseNameString>
4 |  </baseName>
5 |  <baseName>
6 |    <scope>
7 |      <topicRef xlink:href="#processado"/>
8 |    </scope>
9 |    <baseNameString>é processado por</baseNameString>
10|  </baseName>
11|  <baseName>
12|    <scope>
13|      <topicRef xlink:href="#processa"/>
14|    </scope>
15|    <baseNameString>processa</baseNameString>
16|  </baseName>
17|</topic>

```

O tópico acima (identificado por `assoc_doc_comp`) representa um tipo de associação; ele determina o nome da associação (linha 3), além dos nomes que serão disponibilizados durante a navegação do topic map para cada papel de actuação dos membros envolvidos.

Por fim, especifica-se a associação propriamente dita, no elemento **association**. Além do seu tipo, determina-se os membros e seus respectivos papéis de actuação. Por exemplo, na associação abaixo do tipo `assoc_doc_comp` (linha 3), o tópico *documento* tem relacionado a si o papel *processado*, enquanto que o papel *processa* está relacionado com o tópico *computador*.

```

1 |<association>
2 |  <instanceOf>
3 |    <topicRef xlink:href="#assoc_doc_comp"/>
4 |  </instanceOf>
5 |  <member>
6 |    <roleSpec>
7 |      <topicRef xlink:href="#processado"/>
8 |    </roleSpec>
9 |    <topicRef xlink:href="#documento"/>

```

```

10 |     </member>
11 |     <member>
12 |         <roleSpec>
13 |             <topicRef xlink:href="#processa"/>
14 |         </roleSpec>
15 |         <topicRef xlink:href="#computador"/>
16 |     </member>
17 | </association>

```

A partir dessa associação, a associação entre *computador* e *documento* pode ser entendida de duas formas:

- computador **processa** documento;
- documento é **processado** por computador.

### 11.3.4 Definição das ocorrências de cada tópico

O passo final que colmata o processo de criação do topic map referente ao MC é composto da definição das ocorrências dos tópicos que possuem referências nos recursos de informação disponíveis. Esta etapa é fundamental visto que serão estas ocorrências que permitem, a nível do navegador produzido, aceder aos recursos de informação a partir dos conceitos representados no mapa.

Porém, para concretizar esta tarefa é necessário um cuidado especial: cada tipo de recurso que se queira referenciar define a criação de um novo tópico.

No caso de estudo em apreço, os recursos que se utilizaram foram todos documentos em HTML (na íntegra, ou partes) pertencentes aos sumários e material de apoio disponibilizado no website da disciplina, pela equipa docente. Por esta razão, decidiu-se designar este tipo de ocorrência por *URL*, definindo-se assim o tópico abaixo:

```

1 | <topic id="url">
2 |     <baseName>
3 |         <baseNameString>URL (Uniform Resource Locator)</baseNameString>
4 |     </baseName>
5 | </topic>

```

Após a especificação do tipo das ocorrências, a definição das mesmas pode ser feita sem mais dificuldades. Como se exemplifica abaixo, o tópico *Computador* possui uma ocorrência do tipo *URL* (linha 7) que está descrita no documento *Dicionario.htm*<sup>7</sup> (linha 10). A ocorrência é então acrescentada à definição do respectivo tópico.

```

1 | <topic id="computador">
2 |     <baseName>
3 |         <baseNameString>Computador</baseNameString>
4 |     </baseName>

```

<sup>7</sup>O qual contém uma definição sucinta para cada um dos conceitos básicos do conteúdo curricular da disciplina em causa.

```

5 | <occurrence>
6 |   <instanceOf>
7 |     <topicRef xlink:href="#url"/>
8 |   </instanceOf>
9 |   <resourceRef
10 |     xlink:href="http://www.di.uminho.pt/~gepl/IIA04/Dicionario/Dicionario.htm#comp"/>
11 | </occurrence>
12 </topic>

```

Seguindo esta linha sistemática de pensamento, construiu-se manualmente todo o topic map (segundo a especificação XTM) que traduz o *Mapa de Conceitos* fornecido à partida para a disciplina de *Introdução à Informática para Ciências Sociais*, tendo-se obtido uma descrição com um total de 5360 linhas, contendo: 353 tópicos, 122 associações e 55 ocorrências.

## 11.4 Geração do Navegador para o Mapa de Conceitos

Após a criação do topic map, o processo final consiste da geração automática do navegador pretendido (consoante se explicou na introdução ao caso de estudo, na secção anterior). Para o efeito recorreu-se ao Ulisses.

A partir do topic map especificado em XTM – que foi obtido conforme os passos descritos na Secção 11.3 – o Ulisses produziu automaticamente, e com eficiência, o conjunto de páginas HTML que formam o navegador pretendido para o Mapa de Conceitos do caso de estudo descrito. As páginas geradas propiciam, em qualquer browser da Web, uma navegação conceptual fácil através dos conceitos apresentados e dos recursos ligados. A página inicial desta navegação apresenta-se na Figura 11.3.

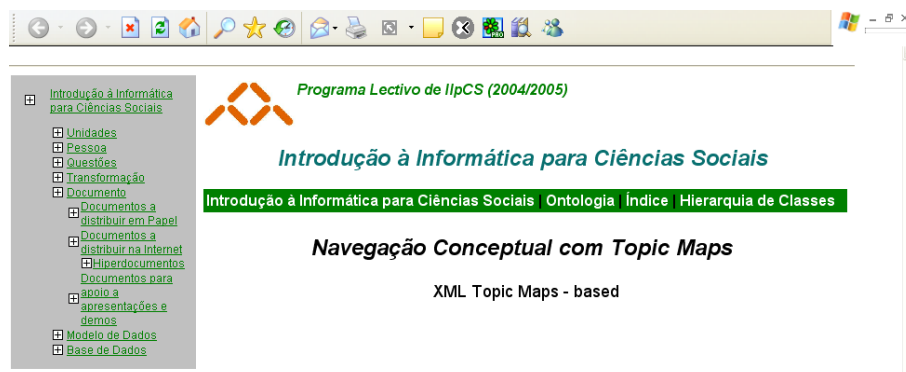


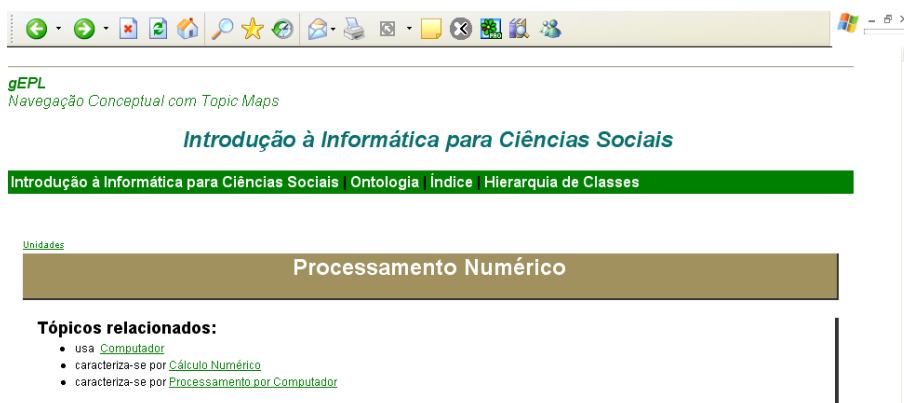
Figura 11.3: Página do inicial do IIPCS gerada pelo Ulisses

A página inicial possui uma representação da hierarquia de tópicos no lado esquerdo da página. Uma das opções é o conceito *Unidades*, referente às unidades lectivas de IIPCS apresentadas na Figura 11.2. A página referente a este conceito está mostrada na Figura 11.4.

A página referente ao conceito *Unidades* possui a indicação de seu tipo (IIPCS) e suas

Figura 11.4: Página referente ao conceito *Unidades*

3 instâncias: *Processamento de Documentos*, *Processamento Numérico* e *Processamento de Informação*. Seguindo uma navegação no grafo representado na Figura 11.2, tem-se a página referente ao conceito *Processamento Numérico* na Figura 11.5.

Figura 11.5: Página referente ao conceito *Processamento Numérico*

A página HTML para o conceito *Processamento Numérico* é composta pelo seu tipo (*Unidades*) e as associações que este conceito possui; neste caso são 3. Uma delas é a associação com o conceito *Computador*, cuja visualização encontra-se na Figura 11.6.

A página referente ao conceito *Computador* (Figura 11.6) possui um diferencial em relação às anteriores: além de ligações Web para o seu tipo e para os tópicos associados (por exemplo, *Hardware* – Figura 11.7), há uma ocorrência que serve para caracterizar o conceito. O conteúdo da URL fornece algum tipo de informação sobre o conceito em questão; no caso, a ligação relaciona o tópico *Computador* com uma definição deste conceito na página desenvolvida para a disciplina de IIPCS.

A página ilustrada pela Figura 11.7 diz respeito ao conceito *Hardware* apresentando, por baixo do tópico, os três grupos de elementos principais que o caracterizam:

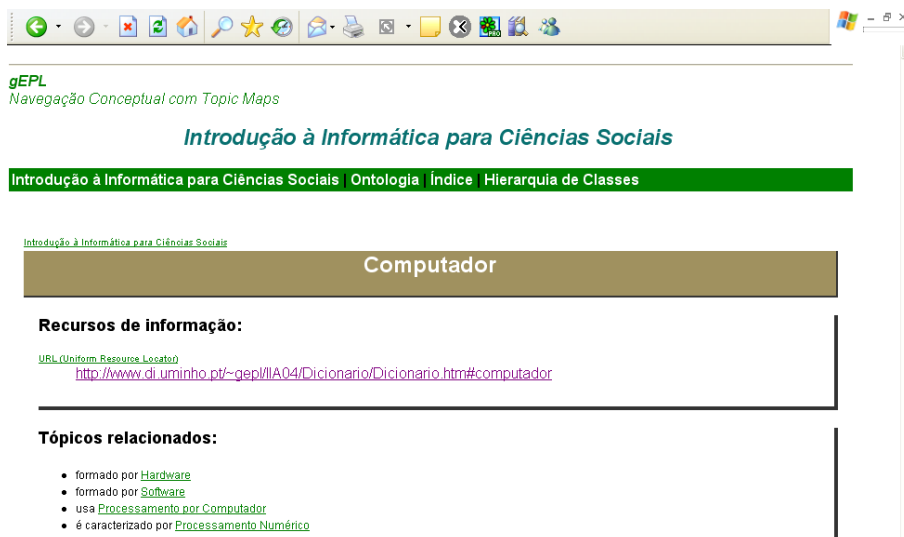


Figura 11.6: Página referente ao conceito *Computador*

**Recursos de Informação:** as ocorrências do tópico, neste caso um recurso de informação acessível através do URL indicado;

**Tópicos Relacionados:** tópicos associados (por coincidência também só um), indicando o papel de actuação (*é parte constituinte de*) de *Hardware* em relação ao tópico *Computador*;

**Instâncias:** as instâncias do tópico; neste caso existem cinco.

Ainda se encontra, na parte superior da página, referência ao tópico do qual *Hardware* é uma instância. Qualquer ligação a um tópico permite aceder directamente à sua própria definição (uma página com a mesma estrutura apresentada na Figura 11.7).

Para além desta informação, específica desse Mapa de Conceitos particular, qualquer página criada pelo Ulisses inclui uma barra de navegação composta por apontadores (*links*) genéricos que permitem aceder: à página inicial do navegador; à ontologia implícita no topic maps (sendo ainda disponibilizadas várias opções de consulta); a um índice alfabético de todos os tópicos; e à visualização gráfica do topic map (neste caso coincidiria com o Mapa de Conceitos).

## 11.5 Sumário do caso de estudo do IipCS

Este capítulo teve dois objectivos principais: (1) a demonstração do processo de geração de Topic Maps a partir de uma ontologia, no caso, representada em Mapas de Conceitos; (2) a utilização do Ulisses para propiciar a navegação conceptual encontrada no topic map gerado.

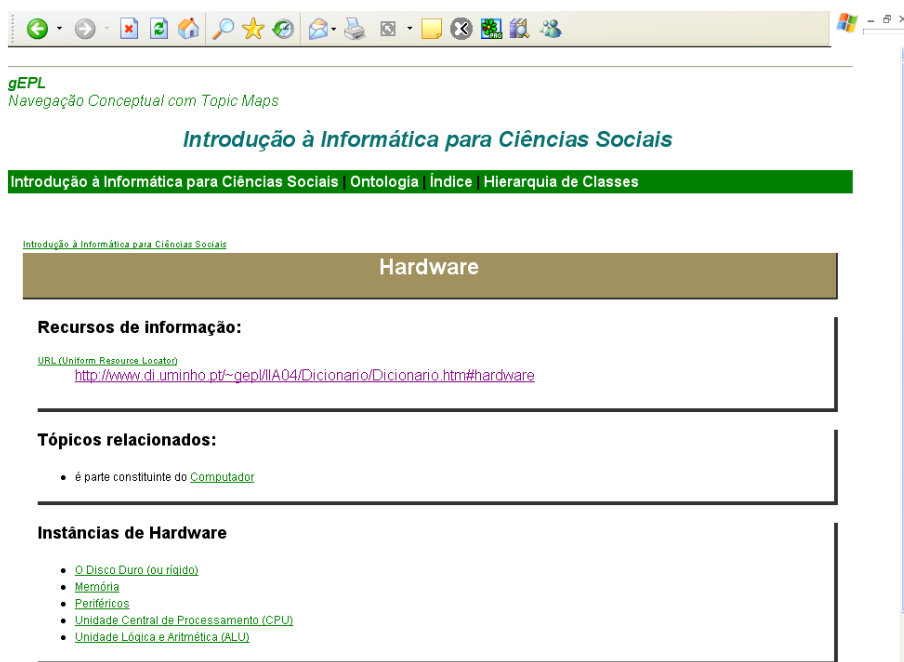


Figura 11.7: Página referente ao conceito *Hardware*

Com o intuito de apetrechar um website de apoio a uma disciplina de Introdução à Informática, com a descrição do seu conteúdo curricular feita através de um Mapa de Conceitos, foi decidido lançar mão aos **Topic Maps** e às tecnologias associadas associadas ao seu processamento. Para isso, estudou-se com cuidado a própria definição de MC (bem semelhante à noção de ontologia), analisou-se a informação nele contida e foram identificados os objectivos que se pretendem atingir ao associar tal instrumento ao dossier de uma disciplina.

Confrontando os resultados dessa análise com o conceito de **Topic Maps**, concluímos que o caminho a seguir, na prossecução do intuito didático acima exposto, passaria pelo uso de norma **Topic Maps**, utilizando a sintaxe XTM para descrever o MC criado.

Procurou-se então sistematizar a tarefa de transcrição dos conceitos e relações do MC para tópicos e associações do **Topic Maps**, como única via para obter rapidamente, com segurança e sem grande custo, a descrição completa do mapa de conceitos da *Introdução à Informática*, fornecido esquematicamente (na forma de um grafo).

Uma vez obtida a especificação XTM, utilizou-se o **Ulisses** para construir mecanicamente o navegador Web pretendido. Os resultados mostram o sucesso da solução tecnológica, pois habilitou-se uma navegação conceptual sobre o MC previamente fornecido.

Contudo, esse caso de estudo mostrou o grau de dificuldade de criação manual de um topic map. O processo de sua edição – composto pela edição individual de cada um dos 353 tópicos e das 122 associações – durou aproximadamente 8 horas; um custo sensivelmente oneroso, portanto, que tende a crescer proporcionalmente ao número de tópicos e associações que devem ser postos no topic map.

A fim de automatizar o processo de geração de **Topic Maps** a partir de fontes heterogéneas de dados, o próximo caso de estudo aplica o **TM-Builder** e o **Oveia** (outros módulos do **Metamorphosis**, do qual faz parte o **Ulisses**) a um caso real: a construção do mapa do site do Departamento de Informática da Universidade do Minho. Também neste caso, o **Ulisses** proporcionaria a geração da navegação a partir de um topic map.



## Capítulo 12

# O TM-Builder e o Oveia para a extracção do mapa do site do DI-UM

*Aqui en la isla el mar,  
y canto mar se sale,  
de sí mismo a cada rato,  
A luz é veloz nos seus olhos de ave  
dice que sí, que no,  
que no, que no, que no, dice que sí,  
en azul, en espuma, en galope,  
dice que no, que no...*

Metáfora, por Pablo Neruda

O presente capítulo apresenta um caso de estudo que demonstra o processo de especificação e extracção de **Topic Maps** com recurso ao **TM-Builder** e ao **Oveia**. No caso em apreço usa-se um **topic map** para representar o mapa do site do Departamento de Informática da Universidade do Minho (DI-UM), descrevendo as componentes presentes nesse site e a navegação conceptual entre elas.

Assim, esse capítulo tem como objectivo principal ilustrar uma aplicação para o **TM-Builder** (Capítulo 7) e o **Oveia** (Capítulo 8) – dois dos módulos do **Metamorphosis** (Capítulo 6). Um outro objectivo é ilustrar o uso de um **topic map** para descrever o mapa de um site.

No primeiro semestre do ano de 2003, foi desenvolvido um novo website para o DI-UM. O objectivo deste website foi tornar todas as informações relativas a este departamento acessíveis, via Web, ao mundo exterior e mesmo internamente aos seus vários colaboradores (docentes, investigadores, funcionários, alunos, etc). Para assegurar a permanente actualidade da informação pretende-se um site dinâmico, pelo que a fonte de informação usada

foi uma base de dados relacional MySQL<sup>1</sup> onde toda a informação a ser disponibilizada estava armazenada.

Uma vez desenhada e implementada a base de dados, o desenvolvimento consistiu na escrita de páginas HTML dinâmicas (nesse caso, em PHP) para efectuar interrogações SQL à base de dados e gerar, dinamicamente, páginas HTML que satisfazem os pedidos de cada utilizador. Este cenário é apresentado na Figura 12.1.

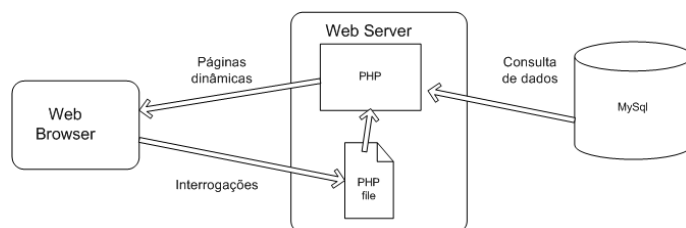


Figura 12.1: Arquitectura do website do DI-UM

Uma vez o website pronto, fez sentido criar um mapa de navegação conceptual, o qual é chamado de *Mapa do Site*. Quando se pensa num mapa de navegação conceptual, está-se a pensar num mapa de conceitos, onde cada conceito pode estar relacionado com os outros e também possui um grupo de recursos associados a si.

Assim, a navegação, em vez de se explicitar em termos de páginas pode ser descrita em termos dos conceitos presentes nessas páginas, ou subconjuntos delas, sendo por isso possível navegar intuitivamente sobre o conhecimento representado no site, acedendo-se então aos recursos associados aos conceitos (as páginas propriamente ditas).

Devido ao facto dos **Topic Maps** se apresentarem como uma boa opção para a representação e navegação sobre o conhecimento de um domínio em particular, o problema abordado neste caso de estudo passou a ser a geração de um topic map para o domínio do DI-UM. Geração essa que se pretendia tão automática quanto possível.

Então, o *Mapa do Site* pretendido será obtido a partir de um topic map gerado com base na especificação da ontologia que descreve os conceitos e relações referentes ao departamento. Para atingir o resultado pretendido, duas abordagens foram empregues:

**Com o TM-Builder:** a base de dados original é descarregada para um documento XML.

A extracção é especificada em XSTM, cujo processamento cria um TM-Builder para este domínio. O TM-Builder, por sua vez, processa o documento XML referente à base de dados e gera um topic map que será guardado num documento no formato XTM;

**Com o Oveia:** a base de dados é interrogada directamente pelo extractor evitando-se descarregamento das informações em XML. A especificação – quer de estrutura da base de dados, quer da ontologia – é realizada nas linguagens XSDS e XS4TM. Neste caso, o Oveia gera como produto final um topic map que será armazenado numa base de dados BD Ontologia, ou num ficheiro XTM.

---

<sup>1</sup><http://www.mysql.com/>

Por curiosidade, refira-se que foi este caso de estudo que nos alertou para a importância de um extractor como o Oveia, pois a primeira abordagem rapidamente conduziu à obsolescência do topic map – cada vez que era produzido uma alteração na base de dados, todo o processo de exportação para XML e de geração do topic map tinha de ser repetido.

Em ambas abordagens, o fruto do processo é um topic map contendo a representação do conhecimento do domínio coberto pelo DI-UM de acordo com a visão do especialista. A partir deste topic map, o Ulisses (Capítulo 10) efectua a criação automática do conjunto de páginas HTML referente ao Mapa do Site desejado. Contudo, no contexto deste capítulo esta questão (geração do navegador com o Ulisses) não será aprofundada para não distrair a atenção do tema central – a extracção de Topic Maps.

Este capítulo inicia com a definição da ontologia para este caso de estudo na Secção 12.1. A aplicação do TM-Builder ao problema apresentado encontra-se na Secção 12.2, enquanto que a aplicação do Oveia ao mesmo problema encontra-se na Secção 12.3. O sumário deste capítulo é realizado na Secção 12.4.

## 12.1 Definição de uma ontologia para o DI-UM

Esta secção apresenta o processo de definição do conhecimento a ser representado sobre o DI-UM. Esse processo foi realizado por um especialista no domínio, um profundo conhecedor não apenas dos dados contidos nos recursos de informação, mas também da estrutura de cada fonte.

Esse processo compreende três etapas: a selecção de quais conceitos devem ser encontrados neste universo de discussão, a definição da taxonomia do DI-UM (relações classe-subclasse) e os relacionamentos diversos entre os conceitos, os quais completam a ontologia em questão.

### 12.1.1 Os conceitos no domínio do DI-UM

No que diz respeito a definição dos temas a serem representados no topic map do DI-UM, a Tabela 12.1 lista 31 conceitos deste domínio, citados em ordem alfabética.

Tabela 12.1: Conceitos encontrados no domínio do Departamento de Informática.

Anos Lectivos	Área de Interesse	Categorias	Conferências
Contactos	Departamento de Informática	Desenvolvimento	Destaques
Disciplinas	Ensino	Equipamentos	Financiadores
Grupos	Investigação	Licenciaturas	Marcações
Membros	Novidades	Palavras Chave	Produtos
Projectos	Publicações	Recursos	Reservas
Salas	Software	Tipo de Curso	Tipo de Novidade
Tipo de Projecto	Tipo de Publicação	Turmas	

Os conceitos apresentados na Tabela 12.1 serão definidos como tipos de tópico nas linguagens de especificação de extracção de topic maps. Assim sendo, cada tópico que é extraído dos recursos de informação será, portanto, uma instância de um destes tipos.

### 12.1.2 Uma taxonomia para o DI-UM

Uma vez definidos os conceitos ou classes deste domínio, constrói-se uma taxonomia que representa a hierarquia entre essas classes.

A visualização gráfica da taxonomia definida é encontrada na Figura 12.2. Por questões pragmáticas, nesta figura é possível observar apenas parte da hierarquia de classes composta pelos conceitos apresentados na Secção 12.1.1.

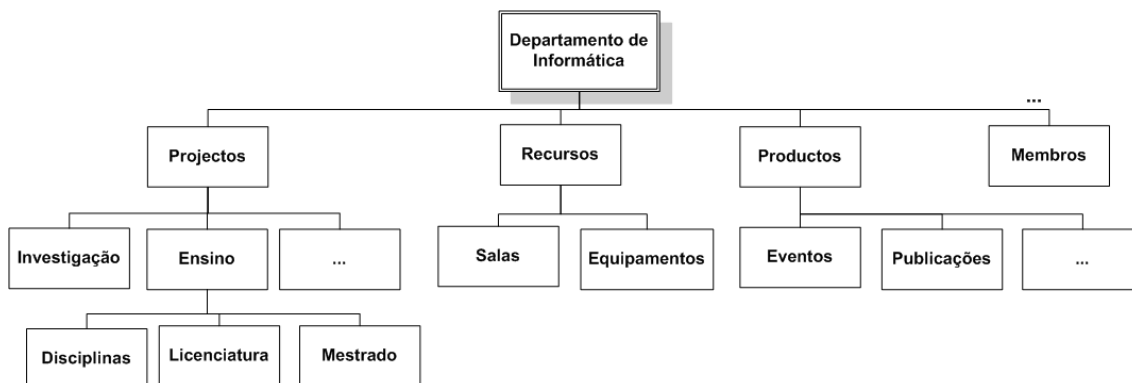


Figura 12.2: Uma taxonomia para o DI-UM

A partir desta taxonomia, o conhecimento sobre o DI-UM passa a estar estruturado. Porém, ainda sente-se a falta de outros relacionamentos, diferentes do tipo classe-subclasse. Esses demais relacionamentos tornarão a representação que está, até o momento em forma de árvore, em uma representação em forma de grafo, conforme apresentado na próxima subsecção.

### 12.1.3 Relacionamentos entre os conceitos da taxonomia

Após a análise detalhada da informação sobre o domínio do DI-UM, chegou-se à uma ontologia que está descrita graficamente na Figura 12.3.

Cada relacionamento envolvendo os conceitos apresentados na Figura 12.3 possui suas características particulares, que são os papéis de actuação que cada membro desempenha. Por exemplo, através da relação entre *grupos* e *pessoas*, é possível nomear a associação (*conter*) e definir os papéis de actuação: *contém* e *pertence*. O grupo contém pessoas; a pessoa pertence ao grupo.

As associações da ontologia deste domínio são basicamente de 4 tipos: *ser um* (por exemplo entre *cursos* e *tipos de curso*), *possuir* (ex: entre *pessoas* e *contactos*), *efectuar* (ex: entre *pessoas* e *marcações*) e *conter* (ex: entre *pessoas* e *grupos*).

Todas as associações entre os conceitos deste domínio possuem papéis de actuação específicos para cada membro participante, como o exemplo apresentado anteriormente. Essas relações definem a semântica que o topic map gerado pelo TM-Builder e pelo Oveia contém.

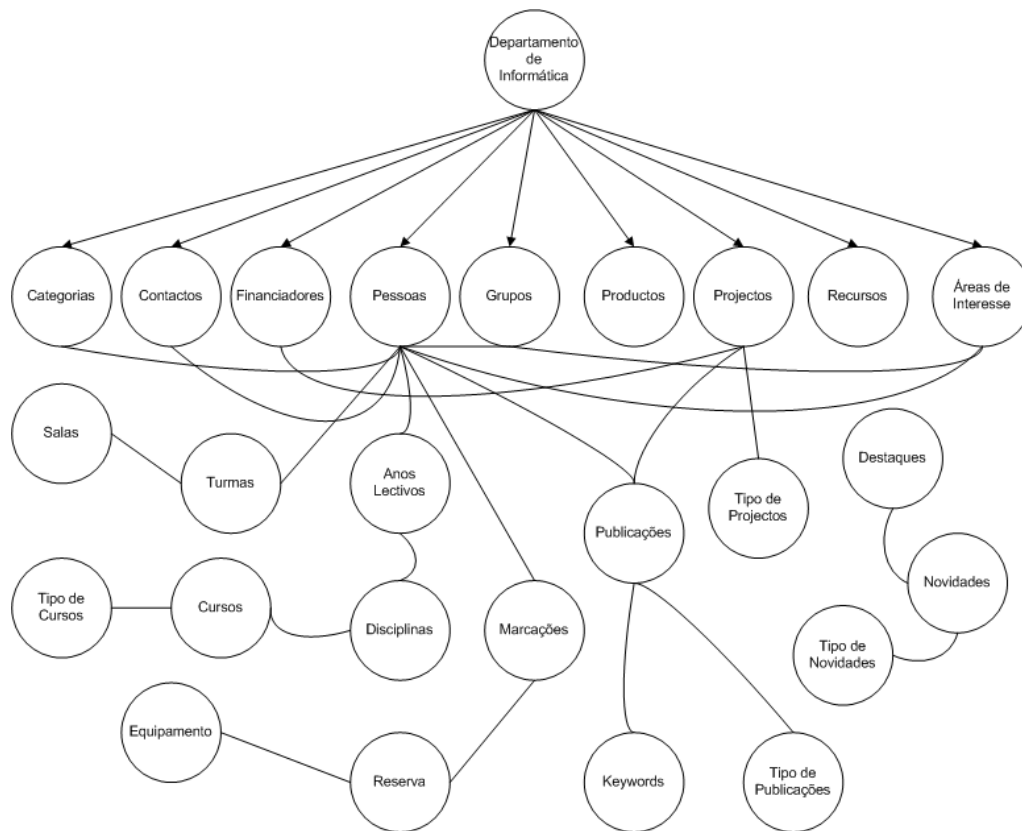


Figura 12.3: A visão gráfica de uma ontologia para o DI-UM

Convém ressaltar que a ontologia aqui definida é apenas uma visão do domínio apresentado que servirá de base para as especificações XSTM e XS4TM que serão definidas nas próximas secções. Contudo, ela pode sofrer alterações conforme o desejo do projectista, ocasionando consequentemente alterações nas especificações de extracção.

## 12.2 TM-Builder aplicado ao DI-UM

Para a obtenção do topic map, que deu origem à primeira versão do *Mapa do Site* do DI-UM, a partir do uso do TM-Builder, seguiram-se os passos abaixo enumerados:

1. Obtenção de um documento XML contendo toda a informação que se encontra nas tabelas da base de dados MySQL do DI-UM. esta fase prévia foi necessária porque o TM-Builder é uma ferramenta que permite a extracção de topic maps exclusivamente a partir de documentos XML;
2. Criação de uma especificação XSTM contendo a definição da ontologia do DI-UM, determinando o processo de extracção do topic map desejado;

3. Processamento da especificação XSTM para a geração do TM-Builder referente a este domínio;
4. Processamento do documento XML obtido a partir da base de dados pelo TM-Builder, gerando como saída, um topic map na sintaxe XTM.

As próximas subsecções mostram como cada um dos passos acima foi efectuado.

### 12.2.1 *Dump* da base de dados para um documento XML

Usando uma das facilidades de um gestor de base de dados, foi possível criar automaticamente um documento XML que continha a estrutura da base de dados e todo o seu conteúdo. Esta exportação podia ser feita tabela-a-tabela ou toda a base de dados de uma única vez, sendo a segunda opção adoptada na situação em causa.

Neste projecto foi usado o MySQL<sup>2</sup> como SGBD (Sistema de Gestão de Base de Dados), tornando simples de executar e eficiente a tarefa de criação do documento XML. Contudo, essa fase não depende da ferramenta usada, na medida em que é uma funcionalidade disponível em quase todos os SGBD actuais.

Nesse processo de criação de um documento XML com o conteúdo da base de dados, o MySQL determina que cada tabela será representada por um elemento com o seu nome. A estrutura deste elemento será formada pela sequência dos campos da tabela. Cada um dos campos das tabelas define a criação de um sub-elemento do elemento referente à tabela onde ele se encontra.

A Figura 12.4 mostra o relacionamento entre as tabelas *Pessoas*, *Grupos*, *Áreas de Interesse* e *Pessoas e Áreas*. Estes relacionamentos – uma parte apenas do DER completo da base de dados do DI-UM – servirão como exemplo para a especificação da extracção de topic maps nas linguagens XSTM e XS4TM.

Um extracto da base de dados contendo um tuplo das tabelas encontradas na Figura 12.4 está abaixo representado:

```

1 <di>
2   <Pessoas>
3     <CodPes>20</CodPes>
4     <Nome>Pedro Manuel Rangel Santos Henriques</Nome>
5     <Apelido>Rangel Santos Henriques</Apelido>
6     <Sigla>PRH</Sigla>
7     <NumMec>187</NumMec>
8     <Email>prh@di.uminho.pt</Email>
9     <Telefone>4459</Telefone>
10    <URL>http://di.umiho.pt/~prh</URL>
11    <AnoInicio>1985</AnoInicio>
12    <MesInicio>10</MesInicio>
13    <DiaInicio>10</DiaInicio>
14    <CodCat>2</CodCat>
15    <CodGrupo>8</CodGrupo>
16  </Pessoas>

```

---

<sup>2</sup><http://www.mysql.com/>

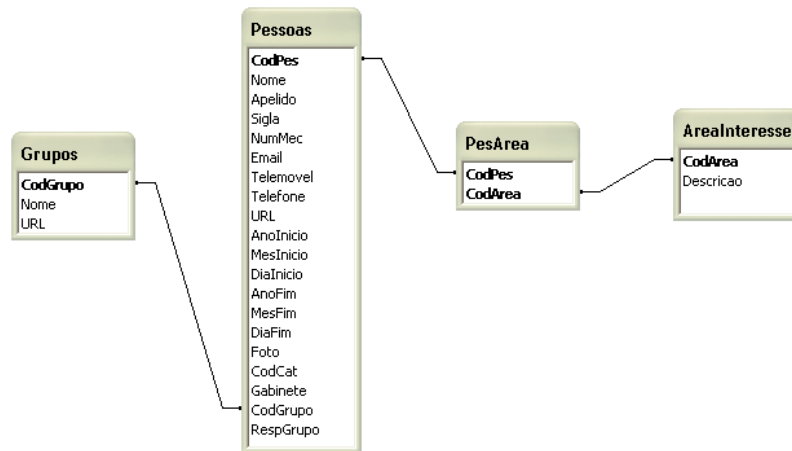


Figura 12.4: Parte dos relacionamentos do DI-UM

```

17 | <Grupos>
18 |   <CodGrupo>8</CodGrupo>
19 |   <Nome>EPL</Nome>
20 | </Grupos>
21 | <AreaInteresse>
22 |   <CodArea>37</CodArea>
23 |   <Descricao>Anotação: XML e Tecnologias Associadas</Descricao>
24 | </AreaInteresse>
25 | <PesArea>
26 |   <CodPes>20</CodPes>
27 |   <CodArea>37</CodArea>
28 | </PesArea>
29 | ...
30 | <di>

```

Percebe-se que a estruturação da informação é simples, não contendo sub-níveis a partir dos elementos referentes aos campos das tabelas. Sendo assim, o XML gerado vai possuir apenas 3 níveis:

1. a raiz;
2. o nível referente às tabelas;
3. o nível referente aos campos da tabela.

No final deste passo, foi obtido um documento XML contendo 10986 linhas referentes a 43 tabelas.

### 12.2.2 Especificação em XSTM

A linguagem XSTM – detalhada no Capítulo 7 – determina o processo de extracção de topic maps a partir de documentos XML. A especificação XSTM para o DI-UM é dividida em 4 etapas, apresentadas nas subsecções a seguir.

## Especificação da Taxonomia do DI-UM

Neste passo, todos os tipos de tópicos devem ser definidos, gerando a hierarquia de classes encontrada neste domínio, apresentada na Secção 12.1.2. Inicialmente, declara-se o tipo de tópico que é a raiz da taxonomia: *Departamento de Informática*. A especificação deste tipo de tópico, de acordo com a sintaxe XSTM, realiza-se no elemento `firstTopicType`:

```

1 <xstm>
2   <topicTypes>
3     <firstTopicType>
4       <id>DI</id>
5       <name>Departamento de Informática</name>
6     </firstTopicType>
7     ...
8   </topicTypes>
9 </xstm>

```

Uma vez definido o tipo de tópico raiz, passa-se aos próximos níveis da taxonomia. Em XSTM, os tipos de tópicos podem ser definidos como instâncias de outros tipos de tópicos. No caso do DI-UM, o *Departamento de Informática* é composto por várias sub-classes, conforme apresentado na Figura 12.2.

Na especificação de cada sub-classe, necessariamente deve conter uma referência ao identificador do tipo de tópico do qual ele é uma instância no elemento `instanceOf`.

Para exemplificar, apresenta-se a especificação dos tipos de tópicos *Membros* e *Projectos* que são instâncias de *Departamento de Informática*, assim como a definição de um terceiro nível na taxonomia, com a introdução do tipo de tópico *Tipos de Projectos*, o qual é uma instância de *Projectos*. Isso pode ser visualizado no segmento de código XSTM abaixo:

```

1 <topicTypes>
2   ...
3   <topicType>
4     <id>Membros</id>
5     <instanceOf>DI</instanceOf>
6     <name>Membros do Departamento de Informática</name>
7   </topicType>
8   <topicType>
9     <id>Projectos</id>
10    <instanceOf>DI</instanceOf>
11    <name>Os Projectos do DI</name>
12  </topicType>
13  <topicType>
14    <id>TipoProj</id>
15    <instanceOf>Projectos</instanceOf>
16    <name>Tipos de Projectos</name>
17  </topicType>
18  ...
19 </topicTypes>

```

Deste modo, o elemento `<topicTypes>` contém a definição completa da taxonomia do DI-UM, pois todas as classes dessa taxonomia devem estar especificadas neste elemento. Cada sub-elemento de `<topicTypes>` determina a criação de um tópico pelo TM-Builder.



## Especificação da extracção dos Tópicos

A definição dos tópicos que serão extraídos do documento XML é definida no elemento `<topics>`. Cada tópico extraído será instância de algum dos tipos de tópicos definidos no passo anterior.

Como forma de exemplificar, a seguir encontra-se a declaração XSTM dos tópicos referentes aos tipos *membros* e *grupos*. Os tópicos do tipo *membros* são extraídos a partir do caminho XPath `//di/Pessoas`, enquanto que os tópicos do tipo *grupos* são extraídos a partir do caminho XPath `//di/Grupos`. A especificação XSTM referente a extracção destes 2 tipos de tópico está abaixo demonstrado:

```

1 <topics>
2   <topic>
3     <xpath id="CodPes" name="Nome">//Pessoas</xpath>
4     <type>Membros</type>
5     <resourceData scope="Email">Email</resourceData>
6     <resourceRef instanceOf="Website">URL</resourceRef>
7   </topic>
8   <topic>
9     <xpath id="CodGrupo" name="Nome">//Grupos</xpath>
10    <type>Grupos</type>
11  </topic>
12  ...
13 </topics>

```

Enquanto que os tópicos do tipo *grupos* (linha 8 a 11) somente serão compostos por um identificador e um nome (linha 9), os tópicos do tipo *membros* (linha 2 a 7) conterão, além de seus nomes e identificadores (linha 3), referências ao *e-mail* (linha 5) e ao *website* (linha 6) de cada pessoa.

O processo visto acima foi realizado para todos os tipos de tópico que possuem instância no documento XML obtido da base de dados do DI-UM. No total, dos 34 tipos definidos em `<topicTypes>`, 26 deles possuem instâncias no documento XML; para cada um destes, foi realizada uma especificação em um elemento `<topic>`. Os demais são tipos de tópico, cujas instâncias são somente outros tipos de tópicos, como o caso do tópico raiz (*Departamento de Informática*).

Cada elemento `<topic>` determina a criação de N tópicos, pois cada especificação irá criar um tópico para cada elemento que existir no caminho XPath definido.

## Especificação dos Tipos de Associação

Como apresentado na Secção 12.1.3, existem várias associações envolvendo os tópicos no domínio do DI-UM. Cada associação deve ter um tipo, o qual definirá os papéis de actuação de cada membro pertencente a este tipo de associação.

De acordo com a ontologia do DI-UM, foram especificados 24 tipos de associações, onde cada tipo é definido no elemento `<assocType>`. Cada definição de um tipo de associação estabelece os papéis de actuação para seus membros.

Para exemplificar, toma-se como exemplo a associação entre os tipos de tópicos *grupos* e *membros*, onde os papéis de actuação desta relação são, respectivamente: *contém o membro* e *pertence ao grupo*.

A especificação do tipo de associação entre *grupos* e *membros* visualiza-se abaixo:

```

1 <assocTypes>
2   <assocType>
3     <id>membros-grupos</id>
4     <name>Membros e Grupos</name>
5     <memberAssoc>
6       <scope>membro-pertence-grupo</scope>
7       <description>pertence ao grupo</description>
8     </memberAssoc>
9     <memberAssoc>
10      <scope>grupo-contem-membro</scope>
11      <description>contém o membro</description>
12    </memberAssoc>
13  </assocType>
14  <assocType>...</assocType>
15</assocTypes>

```

Este processo é repetido para todos os 24 tipos de associações definidos na ontologia do DI-UM. Cada elemento `<assocType>` define a criação de um tópico pelo TM-Builder.

### Especificação da extracção das associações

Após a definição dos tipos de associação, o passo final da especificação XSTM consiste da definição da extracção das associações.

O elemento `<assoc>` permite a especificação de todas as associações que serão extraídas a partir do documento XML fonte. Este elemento possui dois modelos de extracção de associações: as declarações *one2one* e *M2N*.

O elemento *one2one* expressa relacionamentos que podem ser obtidos a partir de algum elemento com conteúdo igual entre os tópicos definidos como membros de uma determinada associação.

Essa situação verifica-se nos relacionamentos entre *Membros* e *Grupos*. Cada grupo pode ser identificado pelo conteúdo do seu sub-elemento *CodGrupo* (caminho XPath `//Grupos/CodGrupo`). Por sua vez, cada pessoa possui uma referência a qual grupo ela pertence, determinado pelo seu sub-elemento também chamado *CodGrupo* (caminho XPath `//Pessoas/CodGrupo`).

Isso significa que toda vez que houver uma similaridade entre o conteúdo destes elementos, uma associação do tipo *membros-grupos* será criada. Por exemplo, de acordo com o segmento de XML apresentado na Subsecção 12.2.1, o membro *Pedro Henriques* está associado ao grupo *EPL*, pois ambos os caminhos XPath possuem o mesmo conteúdo.

Em XSTM, a associação entre os tipos de tópicos *Membros* e *Grupos* é especificada da seguinte forma:

```

1 <assocs>
2   <one2one>
3     <type>membros-grupos</type>
4     <members>
5       <element>
6         <topicAssoc ref="CodGrupo">Membros</topicAssoc>
7         <role>membro-pertence-grupo</role>
8       </element>
9       <element>
10        <topicAssoc ref="CodGrupo">Grupos</topicAssoc>
11        <role>grupo-contem-membro</role>
12      </element>
13    </members>
14  </one2one>
15  ...
16 </assocs>

```

Cada associação do tipo *membros-grupos* será criada quando o conteúdo dos elementos `//Pessoas/CodGrupo` e `//Grupos/CodGrupo` forem iguais. Esses caminhos foram formados pelo processador XSTM-P baseado na concatenação do caminho XPath definido na declaração de extracção do tópico (definido na declaração `//topic/xpath`) com o atributo `//members/element/topicAssoc/@ref` onde este mesmo tópico actua em uma associação. A Tabela 12.2 apresenta esse caso.

Tabela 12.2: Associações *one2one*

Topic	Membros	Grupos
(1) <code>//topic/xpath</code>	<code>//Pessoas</code>	<code>//Grupos</code>
(2) <code>//members/element/topicAssoc/@ref</code>	<code>CodGrupo</code>	<code>CodGrupo</code>
Concatenação: (1)+(2)	<code>//Pessoas/CodGrupo</code>	<code>//Grupos/CodGrupo</code>

No total, são especificadas 16 associações com o elemento `<one2one>`. Além destas, algumas associações necessitam ser especificadas no elemento `<M2N>`, pois não existe um relacionamento directo entre os tópicos envolvidos. Geralmente, isto ocorre quando há um descarregamento de uma base de dados para um documento XML.

Nos casos onde há um relacionamento muitos-para-muitos entre duas ou mais tabelas, tem-se uma tabela de relacionamentos que contém chaves estrangeiras que estão relacionadas com as chaves primárias das tabelas envolvidas.

Este é o caso da associação entre *membros* e *áreas de interesses*. No DER da base de dados do DI-UM, há uma tabela de relacionamento que contém referências às chaves primárias dessas tabelas, pois um membro pode possuir várias áreas de interesse, assim como uma área de interesse pode ser a preferência de vários membros. Essa tabela de relacionamento, chamada *PesArea*, foi apresentada na Figura 12.4; sua estrutura é composta apenas pelas chaves estrangeiras que estão relacionadas com as tabelas *Pessoa* e *AreaInteresse* e é descrita pelo DTD abaixo:

```

1 <!ELEMENT PesArea (CodPes, CodArea)>
2 <!ELEMENT CodPes (#PCDATA)>
3 <!ELEMENT CodArea (#PCDATA)>

```

Em XSTM, a tabela de relacionamento é definida no elemento `elementIndex`. Neste caso em particular, este sub-elemento contém o caminho XPath para a tabela `//PesArea`.

Essa tabela relaciona os membros e as áreas de interesses, pois o sub-elemento *CodPes* é a referência ao identificador dos tópicos do tipo *Membros*, assim como *CodArea* é a referência ao identificador dos tópicos do tipo *Áreas de Interesses*.

O exemplo abaixo apresenta a especificação XSTM para a associação entre membros e áreas de interesse.

```

1 <assocs>
2   ...
3   <M2N>
4     <type>membros-areas</type>
5     <members>
6       <elementIndex>//PesArea</elementIndex>
7       <element attribute="CodPes">
8         <topicAssoc>Membros</topicAssoc>
9         <role>pertence-a-area-de</role>
10      </element>
11      <element attribute="CodArea">
12        <topicAssoc>AreaInteresse</topicAssoc>
13        <role>area-de</role>
14      </element>
15    </members>
16  </all2all>
17  ...
18 </assocs>

```

A Tabela 12.3 mostra como o processador XSTM-P interpreta o código acima. O conteúdo do identificador do tópico membro da associação deve ser igual ao conteúdo de sua referência na tabela de relacionamento.

Tabela 12.3: Associações *MxN*

	Identificador do tópico: //topic/xpath + //topic/xpath/@id	Referência na Tabela de Relacionamento: //members/elementIndex + //members/element/@attribute
Membros	//Pessoas/CodPes	//PesArea/CodPes
Área de Interesse	//AreaInteresse/CodArea	//PesArea/CodArea

Usando como exemplo o documento XML da Subsecção 12.2 e a associação definida acima, percebe-se que *Pedro Rangel Henriques* pertence à área de *Anotação: XML e Tecnologias Associadas*; olhando por outro ponto de vista, pode-se dizer que *Anotação: XML e Tecnologias Associadas* é a área de *Pedro Rangel Henriques*. Os papéis de actuação “*pertence à área de*” e “*é a área de*” identificam os tipos de tópico participantes deste tipo de associação.

### 12.2.3 Processamento e resultados do TM-Builder aplicado ao DI-UM

Após a conclusão da especificação XSTM (construída na subsecção anterior), o próximo passo para a obtenção do topic map referente ao DI-UM foi o processamento desta especificação pelo XSTM-P; deste processo resultou num TM-Builder. Esse TM-Builder processou o XML obtido a partir da exportação da informação da base de dados, produzindo o topic map na sintaxe XTM.

Em termos práticos, o caso de estudo do DI-UM está centralizado em um documento XML com 10986 linhas, contendo toda a estrutura e os dados que estavam, até então, armazenados na base de dados MySQL original em 43 tabelas.

A partir da estrutura deste documento, definiu-se a especificação XSTM com 888 linhas, contendo a especificação de 34 tipos de tópicos e 24 tipos de associações, além da definição de 26 regras de instanciação de tópico e 23 regras de instanciação de associações.

O processador XSTM-Periou, a partir da especificação XSTM em questão, o TM-Builder (uma folha de estilos XSL) com 2442 linhas. O TM-Builder, ao processar o documento XML, criou um topic map com 22175 linhas, contendo 658 tópicos e 942 associações.

Este processo foi realizado em um ambiente composto por um microcomputador Intel Celeron, com 2.6GHz, 512MB de memória RAM e 5GB livres no disco rígido, utilizando o processador XML *Saxon 6.5.2*. O tempo de criação do TM-Builder foi de 2.1 segundos, enquanto que o processamento do TM-Builder, tomando como entrada o XML do DI-UM, durou 1.1 segundos para efectuar a criação do topic map completo.

Foi também testado a extracção do topic map com outros documentos XML que possuíam a mesma estrutura do documento obtido a partir do descarregamento da base de dados.

Inicialmente, a base de dados não estava completamente preenchida (com todas os dados referentes ao DI-UM); contudo, sua estrutura era a mesma encontrada na base de dados actual. Usou-se esta base de dados incompleta (em termos de conteúdo) para verificar o comportamento do TM-Builder em relação a outra fonte de dado com a mesma estrutura. A partir dela, foi obtido um documento XML com 2834 (25,79% do tamanho do XML originalmente obtido da base de dados) linhas que, ao ser processado pelo mesmo TM-Builder, criou um topic map com 6932 linhas (31,26% do topic map completo), contendo 297 tópicos (45,13%) e 192 associações (20,38%). Este processamento, realizado no mesmo ambiente acima citado, teve a duração de 0.3 segundos (27,27% do tempo tomado pela extracção anterior).

A partir destes dados é possível concluir que o TM-Builder obtido a partir da especificação XSTM descrita na Subsecção 12.2.2 foi capaz de processar dois documentos XML que pertencem ao mesmo esquema, ou seja, não foi necessário adaptações nessa especificação.

Os tempos obtidos para a extracção dos topic maps em ambos os casos são proporcionais: conforme se aumenta o tamanho da fonte XML, incrementa-se o tempo necessário (baseado numa progressão aritmética) para o TM-Builder percorrer a fonte para a geração dos tópicos e associações.

## 12.3 Oveia aplicado ao DI-UM

Ao contrário do TM-Builder, o Oveia permite a extracção de topic maps directamente de fontes heterogêneas de informação. Isso faz com que a base de dados relacional que contém toda a informação sobre o DI-UM pudesse ser utilizada directamente neste caso, sem a necessidade de proceder, previamente, ao seu descarregamento para XML.

Os passos de execução no processo de criação do topic map, que deu origem à segunda versão do Mapa do Site do DI-UM realizado pelo Oveia estão abaixo enumerados:

1. Especificação XSDS das fontes de informação a serem utilizadas na extracção;
2. Especificação XS4TM do topic map a ser extraído;
3. Processamento do Oveia, tomando como entrada as especificações acima e as fontes de informação citadas em XSDS;
4. Geração do topic map, permitindo a escolha do formato de armazenamento: BD Ontologia ou um documento na sintaxe XTM.

Nas próximas subsecções, é mostrado como cada passo foi efectuado.

### 12.3.1 Especificação em XSDS

Devido ao facto de se ter apenas uma fonte de informação neste caso de estudo (a base de dados do DI-UM), será definido um único **datasource**, representando a própria base de dados MySQL.

Na definição deste **datasource** é especificado no atributo **extratorDriver** qual **driver** de extracção do Oveia será utilizado<sup>3</sup>. Além disso, é definido o identificador o nome desta fonte de dados: *BD\_DI*.

Como parâmetros, define-se a URL de conexão com a base de dados (**connectionURL**), o utilizador e a palavra-chave (**user**, **password**) e o driver de conexão JDBC (**jdbcDriver**).

A especificação XSDS está abaixo demonstrada:

```

1 <resources>
2   <datasources>
3     <datasource extratorDriver="br.uneb.dcet.tmbuilder.drivers.DataBase" name="BD_DI">
4       <parameter name="connectionURL">jdbc:mysql://localhost/uminho_di</parameter>
5       <parameter name="user">root</parameter>
6       <parameter name="password"/>
7       <parameter name="jdbcDriver">org.gjt.mm.mysql.Driver</parameter>
8     </datasource>
9   </datasources>
10  <datasets>
11    <dataset name="Membros" database="BD_DI">
12      SELECT CodPes, Nome, Apelido, Email, Telefone, URL, CodCat, CodGrupo FROM pessoas
13    </dataset>
14    <dataset name="Grupo" database="BD_DI">
15      SELECT CodGrupo, URL FROM Categorias
16    </dataset>
17    <dataset name="AreaInteresse" database="BD_DI">
18      SELECT CodArea, Descricao FROM Cursos
19    </dataset>
20    <dataset name="PesArea" database="BD_DI">
21      SELECT CodPes, CodArea FROM Cursos
22    </dataset>

```

<sup>3</sup>Neste caso, o **driver** é: `br.uneb.dcet.tmbuilder.drivers.DataBase`

```

23 | ...
24 | </datasets>
25 | </resources>

```

Na especificação dos **datasets**, usou-se a linguagem de interrogação SQL devido ao facto da fonte de informação ser uma base de dados relacional.

No exemplo acima demonstrou-se a definição dos **datasets** para as tabelas apresentadas na Figura 12.4. Foram seleccionados os campos necessários para a definição do topic map, de acordo com o projecto do especialista no domínio, além de definir um nome para cada **dataset** a ser formado a partir da fonte de dados chamada *BD\_DI*.

### 12.3.2 Especificação em XS4TM

Nesta secção é apresentado a especificação em XS4TM para a extracção do topic map sobre o domínio do DI-UM. Esse processo de especificação divide-se em duas etapas, conforme descrito no Capítulo 8: a definição da ontologia do topic map e a definição das instâncias a serem extraídas. Estas etapas estão detalhadas nas próximas subsecções.

#### Especificação da ontologia do topic map a ser gerado

A definição de todos os tipos de tópicos encontrados na taxonomia deste domínio será realizada no elemento **<ontologies>**. Além disso, este elemento permite também a especificação dos tipos de associações.

Em XS4TM, inicialmente define-se os tipos de tópicos. Seguindo o esquema da linguagem XSTM, a declaração do tipo de tópico *Departamento de Informática* é mostrada abaixo, definindo seu identificador e seu nome.

```

1 | <?xml version="1.0" encoding="ISO-8859-1"?>
2 | <xs4tm xmlns="http://www.topicmaps.org/xtm/1.0/" xmlns:xlink="http://www.w3.org/1999/xlink">
3 |   <ontologies>
4 |     <topic id="DI">
5 |       <baseName>
6 |         <baseNameString>Departamento de Informática</baseNameString>
7 |       </baseName>
8 |     </topic>
9 |     ...
10 |   </ontologies>
11 |   ...
12 | </xs4tm>

```

Para a definição de relações classe-subclasse, faz-se uma referência ao identificador do seu tipo no elemento **<instanceOf>**. O tipo de tópico *membros*, que é uma instância de *Departamento de Informática*, tem a seguinte especificação:

```

1 | <topic id="Membros">
2 |   <instanceOf>
3 |     <topicRef xlink:href="#DI"/>

```

```

4 |     </instanceOf>
5 |     <baseName>
6 |         <baseNameString>Membros do Departamento de Informática</baseNameString>
7 |     </baseName>
8 | </topic>

```

Para a especificação do tipo de associação entre *membros* e *grupos*, define-se um tópico como o abaixo visualizado:

```

1 | <topic id="membros-grupos">
2 |   <baseName>
3 |     <baseNameString>Membros e Grupos</baseNameString>
4 |   </baseName>
5 |   <baseName>
6 |     <scope>
7 |       <topicRef xlink:href="#membro-pertence-grupo"/>
8 |     </scope>
9 |     <baseNameString>pertence ao grupo</baseNameString>
10 |   </baseName>
11 |   <baseName>
12 |     <scope>
13 |       <topicRef xlink:href="#grupo-contem-membro"/>
14 |     </scope>
15 |     <baseNameString>contém o membro</baseNameString>
16 |   </baseName>
17 | </topic>

```

Todos os tópicos definidos no elemento `<ontologies>` serão mapeados directamente para o topic map a ser gerado. Portanto, faz-se uma cópia do que foi definido neste elemento para a saída do Oveia. Contudo, os tópicos aqui definidos são importantes para a definição das instâncias a serem extraídas, o que será apresentado na próxima subsecção.

### Especificação da extracção das instâncias dos tópicos definidos na ontologia

O elemento `<instances>` permite a definição da extracção dos tópicos e associações que são instâncias de algum dos tópicos definidos em `<ontologies>`.

A definição dos tópicos a serem extraídos é realizada no elemento `<topic>`, o qual possui a mesma estrutura do elemento `<topic>` da sintaxe XTM, com o acréscimo do atributo `dataset`, o qual é responsável por fazer referencia ao `dataset` (definido na especificação XSDS) que contém a informação desejada para a criação do tópico em questão.

Abaixo, é encontrado a declaração XS4TM para a definição da extracção das instâncias do tipo de tópico **Membros**.

```

1 | <?xml version="1.0" encoding="ISO-8859-1"?>
2 | <xs4tm xmlns="http://www.topicmaps.org/xtm/1.0/" xmlns:xlink="http://www.w3.org/1999/xlink">
3 |   <ontologies>
4 |     ...
5 |   </ontologies>
6 |   <instances>
7 |     <topic dataset="Membros" id="@Membros.CodPes">
8 |       <instanceOf>
9 |         <topicRef xlink:href="#Membros"/>

```



```

10     </instanceOf>
11     <baseName>
12       <baseNameString>@Membros.Nome</baseNameString>
13     </baseName>
14     <occurrence>
15       <instanceOf>
16         <topicRef xlink:href="#Website"/>
17       </instanceOf>
18       <resourceRef xlink:href="@Membros.URL"/>
19     </occurrence>
20     <occurrence>
21       <scope>
22         <topicRef xlink:href="#E-Mail"/>
23       </scope>
24       <resourceData>@Membros.Email</resourceData>
25     </occurrence>
26   </topic>
27   ...
28 </instances>
29 </xsttm>

```

Tal como a especificação XSTM para a extracção dos tópicos *Membros*, os tópicos deste tipo serão compostos por um identificador, um nome e duas ocorrências (uma referência a um recurso externo e um recurso de dados). Estes dados serão extraídos do **dataset** *Membros*, definido em XSDS. O mesmo processo é realizado para todos os tópicos a serem extraídos do recurso de informação.

A etapa seguinte é a definição das associações entre os tópicos. Elas são declaradas no elemento **<association>**, o qual segue a mesma sintaxe que seu homónimo em XTM, com o acréscimo do atributo **dataset**, o qual define de onde vem os dados para sua criação.

O exemplo abaixo ilustrado apresenta a especificação da associação entre *membros* e *grupos*. O **dataset** *Membros* é citado no atributo **dataset** pois nele há uma referência ao tipo de grupo de cada pessoa (em *@Membros.CodGrupo*).

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <xsttm xmlns="http://www.topicmaps.org/xtm/1.0/" xmlns:xlink="http://www.w3.org/1999/xlink">
3   <ontologies>
4     ...
5   </ontologies>
6   <instances>
7     <association dataset="Membros">
8       <instanceOf>
9         <topicRef xlink:href="#membros-grupos"/>
10      </instanceOf>
11      <member>
12        <roleSpec>
13          <topicRef xlink:href="#grupo-contem-membro"/>
14        </roleSpec>
15        <topicRef xlink:href="@Grupos.CodGrupo"/>
16      </member>
17      <member>
18        <roleSpec>
19          <topicRef xlink:href="#membro-pertence-grupo"/>
20        </roleSpec>
21        <topicRef xlink:href="@Membros.CodGrupo"/>
22      </member>
23    </association>
24    ...

```

```

25 | </instances>
26 | </xs4tm>

```

Toda vez que os conteúdos dos elementos seleccionados (no caso `@Grupos.CodGrupo` e `@Pessoas.CodGrupo`) forem idênticos, uma associação será criada, envolvendo os tópicos que possuem esta característica comum.

### 12.3.3 Processamento e resultados do Oveia aplicado ao DI-UM

A obtenção do topic map de acordo com a base de dados origem é atingida após o processamento das especificações XSDS e XS4TM, mencionadas nas subsecções anteriores.

Em termos práticos, o caso de estudo do DI-UM está suportado em uma base de dados MySQL com 43 tabelas, representando membros, grupos, áreas de interesse, documentos, projectos, disciplinas, cursos, etc. A partir desta fonte de dados, definiu-se uma especificação XSDS com a informação a ser extraída da fonte de dados e uma especificação XS4TM com a ontologia a ser criada.

A especificação XSDS contém 115 linhas, compreende a definição de 1 `datasource` (a própria base de dados do DI-UM) e de 33 `datasets`, os quais descrevem os dados que devem ser extraídos do recurso de informação.

Por sua vez, a especificação XS4TM possui 1572 linhas, computando um total de 109 definições de tópicos, além da definição de 26 regras de instanciação de tópicos e 23 regras de instanciação de associações.

A BD Ontologia gerada contém o topic map armazenado numa base de dados SQL Server composta por 30 tabelas; ao efectuar a conversão para XTM, são criados 1600 elementos, sendo 658 tópicos e 942 associações (números exactamente iguais aos encontrados no topic map extraído pelo TM-Builder), num ficheiro XTM com precisamente 22175 linhas. Ou seja, o topic map gerado pelo Oveia é similar ao gerado pelo TM-Builder. A diferença entre ambos é apenas a ordenação física dos tópicos e associações, o que não implica diferença alguma na rede semântica do topic map.

### Testes do Oveia

Assim como o TM-Builder, este processo foi realizado num ambiente composto por um microcomputador Intel Celeron, com 2.6GHz, 512MB de memória RAM e 5GB de espaço livre no disco rígido. Todos os serviços (a base de dados MySQL origem, o Oveia e a base de dados SQL Server onde foi armazenado a BD Ontologia) foram executados no mesmo ambiente. O Oveia utilizou a Máquina Virtual Java da Sun Microsystems (JVM)<sup>4</sup>. A tarefa de extracção dos dados do `datasource`, construção do topic em memória e armazenamento do mesmo na BD Ontologia foi concluída no tempo de 45 segundos.

Após a extracção do topic map para o DI-UM, foram realizados testes de carga com dados fictícios baseados na mesma especificação de extracção; ou seja, a partir da mesma

<sup>4</sup><http://java.sun.com/downloads/>

especificação foi processado a extração com uma base de dados contendo uma quantidade de dados superior. O teste foi realizado no mesmo ambiente descrito na subsecção anterior. O tempo médio para a extração de um elemento (tópico ou associação) foi de 0,028125 segundo.

Na Figura 12.5 é apresentado o gráfico de tempo de execução para a extração de elementos do topic map, para dados fictícios colocados na base de dados origem. Para este teste foram extraídos 862, 3880, 8116 e 16574 elementos respectivamente.

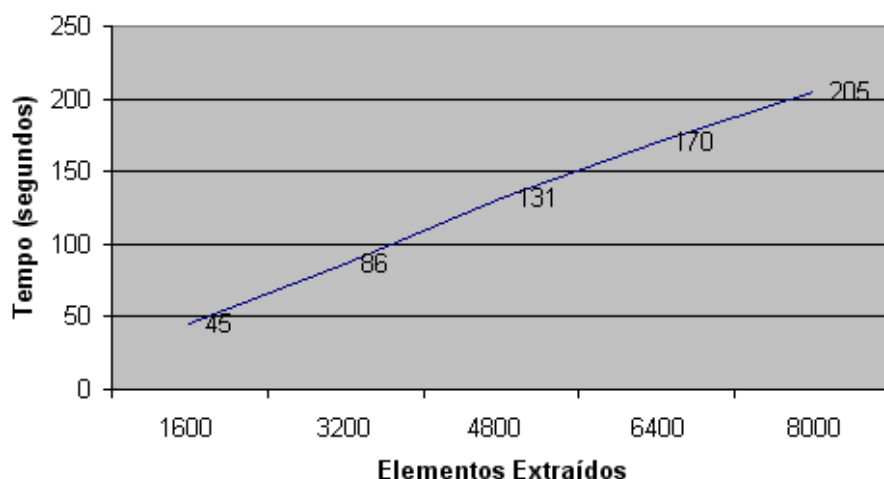


Figura 12.5: Avaliação de Tempo de Extração

Os resultados, em termos de tempo de extração encontram-se na Tabela 12.4.

Tabela 12.4: Teste de desempenho do Oveia

Caso	Elementos extraídos	Tempo de extração total (segundos)	Tempo por elemento (segundos)
(1)	1600	45	0,028125
(2)	3200	86	0,026875
(3)	4800	131	0,027291
(4)	6400	170	0,026562
(5)	8000	205	0,025625

A partir dessa tabela, percebe-se que à medida que o número de elementos extraídos aumenta, a média de tempo de extração por elemento se mantém estável. Isso acontece porque a estruturação da fonte de informação não foi alterada, pois apenas acrescentou-se novos dados à estrutura existente.

## 12.4 Sumário do caso de estudo do DI-UM

Ao longo deste capítulo descreveu-se, em detalhe, duas abordagens à construção de Topic Maps baseados numa ontologia definida sobre o domínio do DI-UM. Os Topic Maps são construídos, a partir de dados extraídos de uma base de dados, pelo TM-Builder e pelo Oveia, demonstrando como cada ferramenta aborda o problema em questão.



Figura 12.6: Visualização do topic map do DI no Ulisses

No caso do TM-Builder, a partir de um documento XML extraído dessa base de dados, obteve-se automaticamente um topic map – escrito de acordo com a sintaxe XTM – a partir de uma especificação XSTM, onde se define a ontologia que deve ser extraída do domínio em questão.

O Oveia, por sua vez, dispensa a etapa de descarregamento da informação da base de dados para XML, pois permite usá-la directamente como fonte de informação. A partir da especificação da fonte e dos dados a serem extraídos (linguagem XSDS) e da ontologia a ser encontrada no topic map (linguagem XS4TM), obtém-se um topic map similar ao gerado pelo TM-Builder.

Com o topic map gerado (seja pelo TM-Builder, seja pelo Oveia), foi possível obter um website que permite a navegação conceptual nesta ontologia. Este website, criado pelo Ulisses, compõe-se de um conjunto de páginas, onde se encontram todos os tópicos da ontologia representada no topic map.

Na Figura 12.6, é ilustrado o aspecto inicial do navegador proporcionado pelo Ulisses, referente ao topic map armazenado na BD Ontologia. Este navegador fornece o total acesso ao topic map extraído, permitindo a navegação através dos conceitos do domínio.

No site gerado pela Variante III do Ulisses (Secção 10.2.3) apresenta-se, na parte superior da página, uma opção para escolher o topic map que se quer visualizar, dentre os armazenados na BD Ontologia. Logo a seguir, o nome da base de dados se apresenta (*UMINHO DI*). No lado esquerdo do corpo da página encontra-se o nome do tópico em questão (*Departamento de Informática da Universidade do Minho*); já no lado direito, se

apresentam as instâncias deste tópico (neste caso 7). Abaixo das instâncias, as associações no qual este tópico faz parte são declaradas; contudo, como esse tópico não participa de qualquer associação, este campo aparece vazio.

Em termos de comparação das abordagens, o processamento do TM-Builder foi mais o rápido, pois se compõe apenas de transformações XML. Contudo, isso aconteceu principalmente devido ao facto de se ter apenas uma fonte de dados (a base de dados MySQL). Se existissem duas ou mais fontes, o TM-Builder teria que tratar individualmente cada fonte para, no fim de tudo, fazer uma fusão entre os topic maps gerados. Por sua vez, o Oveia pode manipular várias fontes ao mesmo tempo e permite a especificação da extracção do topic map em XS4TM, a qual é uma linguagem de especificação de extracção de Topic Maps mais completa do que a linguagem XSTM, permitindo a definição de todas as características encontradas na norma ISO 13250.



## Capítulo 13

# A linguagem XTche numa aplicação de E-Commerce

*Amor?  
Receios, desejos,  
Promessas de paraíso  
Depois sonhos, depois risos  
Depois beijos!  
Depois...  
E depois, Amada?  
Depois dores sem remédio  
Depois pranto, depois tédio Depois... NADA!"*

Minotti Del Picchia

A fim de demonstrar a aplicabilidade do validador semântico de Topic Maps, XTche e a sua conformidade com a norma, este capítulo discute o caso de estudo proposto pela ISO para ilustrar parte dos requisitos contemplados pela TMCL (Nishikawa, Moore, and Bogachev, 2004) e suas instâncias.

O caso proposto pela ISO consiste na validação de um topic map que representa o domínio de conhecimento relacionado com uma empresa de comércio electrónico. A *E-Sell Corporation* (doravante denominada simplesmente por *E-Sell*) é uma empresa de vendas que mantém todo seu conjunto de informação armazenado em base de dados. Para fornecer um acesso unificado aos dados, a *E-Sell* usa topic maps construídos a partir do mapeamento do conteúdo das várias base de dados – as quais possuem distintos esquemas.

Para garantir a consistência dos topic maps gerados, os mesmos devem ser validados seguindo um conjunto de regras. Este capítulo apresenta como XTche permite a definição de tais regras e processa os topic maps, validando-os.

A Secção 13.1 apresenta uma das abordagens possíveis para armazenar a informação sobre

os clientes e produtos<sup>1</sup>, através do uso de **Topic Maps**. O objectivo dessa secção é mapear o domínio da *E-Sell* para um topic map e representá-lo na sintaxe XTM.

A Secção 13.2 lista todas as restrições impostas pelo caso de estudo proposto nos requisitos de TMCL. A Secção 13.3 apresenta a abordagem utilizada por XTche para resolver os problemas citados na Secção 13.2.

A Secção 13.4 mostra o processamento das regras definidas e os resultados obtidos, enquanto que as conclusões sobre o caso de estudo são apresentadas na Secção 13.5.

### 13.1 Modelo de dados da *E-Sell*

Actualmente, aplicações de comércio electrónico têm sido utilizadas amplamente em diversos ramos comerciais. Boa parte das companhias estão utilizando este tipo de aplicação para promover suas vendas, pois os dados dos clientes e dos produtos necessitam ser armazenados electronicamente, de modo que possam ser acedidos através da internet.

Uma solução típica para a gestão desta informação é uma base de dados relacional. O diagrama entidade-relacionamento da Figura 13.1 mostra como a informação é representada na base de dados da *E-Sell*. Esse diagrama é baseado no diagrama encontrado na definição do caso de aplicação de TMCL (Nishikawa, Moore, and Bogachev, 2004). Foram, contudo, acrescentado novos itens à estrutura de modo a cobrir novos casos de restrições, não previstos no caso original.

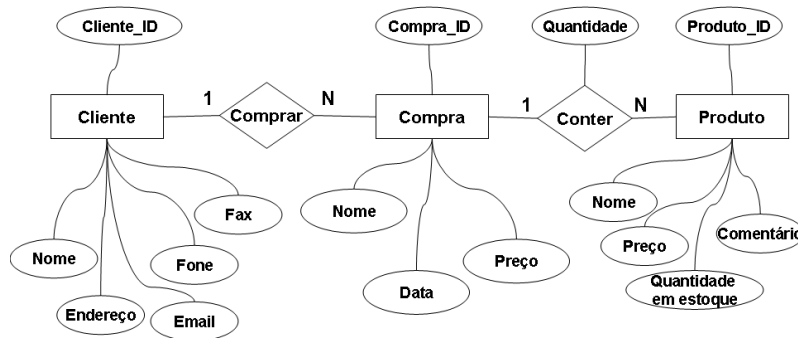


Figura 13.1: Diagrama entidade-relacionamento da base de dados da *E-Sell*

A informação referente à *E-Sell* está contida em 3 tabelas (*clientes*, *ordens de compra* e *produtos*), as quais estão relacionadas entre si através das relações *comprar* e *conter*.

Esse diagrama entidade-relacionamento representa a estrutura na qual os dados da *E-Sell* estão armazenados. A partir dele, define-se o seu mapeamento para um topic map, conforme será apresentado nas próximas subsecções.

<sup>1</sup>Inclui-se também os dados sobre as ordens de compra realizadas pelos clientes.



### 13.1.1 A entidade *produto*

Cada produto encontrado na base de dados será, no topic map, um tópico instância do tópico *Produto*. A informação básica de cada produto compõe-se de um identificador, um nome, a quantidade em estoque e o preço unitário.

O identificador do produto é representado por um identificador de tema. O nome descreve o produto de uma forma que não exista ambiguidade entre os produtos.

A quantidade de cada produto em estoque e o seu preço unitário são armazenados como ocorrências. O preço pode ser representado, inclusive, em euros ou dólares.

Para se adicionar alguma informação extra ao produto, os produtos podem ter ainda uma ocorrência do tipo comentário; ou seja, esse comentário é opcional. Essa característica não estava prevista no caso de estudo de TMCL.

A seguir, apresenta-se um exemplo de um tópico *leitor de DVD* (uma instância do tipo *produto*):

```

1      <topic id="produto">
2          <baseName>
3              <baseNameString>Produto</baseNameString>
4          </baseName>
5      </topic>
6      <topic id="dvd">
7          <instanceOf>
8              <topicRef xlink:href="#produto"/>
9          </instanceOf>
10         <subjectIdentity>
11             <subjectIndicatorRef xlink:href="http://produtos.com/dvd"/>
12         </subjectIdentity>
13         <baseName>
14             <baseNameString>Leitor de DVD</baseNameString>
15         </baseName>
16         <occurrence>
17             <instanceOf>
18                 <topicRef xlink:href="#quantidade"/>
19             </instanceOf>
20             <resourceData>50</resourceData>
21         </occurrence>
22         <occurrence>
23             <instanceOf>
24                 <topicRef xlink:href="#preco"/>
25             </instanceOf>
26             <scope>
27                 <topicRef xlink:href="#euro"/>
28             </scope>
29             <resourceData>85</resourceData>
30         </occurrence>
31     </topic>

```

O tópico *leitor de DVD* (linhas 6 a 31) – uma instância do tópico *produto* (linhas 1 a 5) – possui um indicador de tema (linha 10 a 12), um nome (linha 13 a 15), uma ocorrência do tipo *quantidade* – contendo uma informação textual indicando o valor “50” (linha 16 a 20) – e uma ocorrência do tipo *preço* no contexto *euro* (linha 22 a 30), afirmando que o preço do leitor de DVD é 85 euros.

Percebe-se então que a relação entre os tópicos *produto* e *dvd* é do tipo classe/instância. Contudo, as características estruturais destes tópicos são independentes: cada tópico (ou tópicos de um mesmo tipo) possui sua própria estrutura. Em XTM, não é possível definir que um tópico do tipo *produto* deva ser composto por 2 ocorrências. Para garantir que suas instâncias obedeçam estas regras, uma condição XTche deverá ser imposta.

Todos os demais tópicos instâncias de *produto* possuem uma estruturação similar ao encontrado no tópico *leitor de DVD*, podendo inclusive possuir uma ocorrência do tipo comentário (o que não se apresenta no tópico acima).

### 13.1.2 A entidade *cliente*

Cada cliente é representado por um tópico do tipo *cliente*. Os seus atributos podem ser representados como características de tópico: o nome do cliente é representado por um nome, o endereço e o e-mail como ocorrências. Além disso, os clientes podem possuir um identificador representado por um identificador de tema.

Um cliente pode possuir também um número para contacto, o qual pode ser um número de telefone ou de fax. Um exemplo de um tópico do tipo *cliente* está abaixo representado:

```

1  <topic id="cliente">
2    <baseName>
3      <baseNameString>Cliente</baseNameString>
4    </baseName>
5  </topic>
6  <topic id="grl">
7    <instanceOf>
8      <topicRef xlink:href="#cliente"/>
9    </instanceOf>
10   <subjectIdentity>
11     <subjectIndicatorRef xlink:href="http://www.di.uminho.pt/membros.htm#grl"/>
12   </subjectIdentity>
13   <baseName>
14     <baseNameString>Giovani Librelotto</baseNameString>
15   </baseName>
16   <occurrence>
17     <instanceOf>
18       <topicRef xlink:href="#address"/>
19     </instanceOf>
20     <scope>
21       <topicRef xlink:href="#Brasil"/>
22     </scope>
23     <resourceData>Fortaleza dos Valos - RS</resourceData>
24   </occurrence>
25   <occurrence>
26     <instanceOf>
27       <topicRef xlink:href="#email"/>
28     </instanceOf>
29     <resourceRef xlink:href="mailto:grl@di.uminho.pt"/>
30   </occurrence>
31   <occurrence>
32     <instanceOf>
33       <topicRef xlink:href="#phone"/>
34     </instanceOf>
35     <resourceData>+351 919222367</resourceData>
36   </occurrence>
37 </topic>

```

O tópico *Giovani Librelotto* (linha 6 a 37) – instância de *cliente* (linha 1 a 5) – possui um indicador de tema (linha 10 a 12), um nome (linha 13 a 15), uma ocorrência textual do tipo *endereço* que pertence ao contexto *Brasil* (linha 16 a 24), uma ocorrência do tipo *email* (linha 25 a 30) e uma ocorrência do tipo *Telefone* (linha 31 a 36). Os demais tópicos instâncias de *cliente* seguem a estrutura aqui apresentada.

### 13.1.3 A entidade *ordem de compra*

Quando um cliente em particular deseja efectuar a ordem de compra de algum produto, representa-se essa ordem como um tópico. A ordem conterá os seguintes dados: o identificador único da ordem de compra, a data de compra e o preço total da ordem.

É conveniente ter um nome com o identificador da ordem de compra, para fins de apresentação em um navegador. A data da ordem e o preço da ordem representam ocorrências.

Abaixo, tem-se um exemplo de um tópico do tipo *ordem de compra*, contendo suas características principais:

```

1      <topic id="ordem-de-compra">
2          <baseName>
3              <baseNameString>Ordem de Compra</baseNameString>
4          </baseName>
5      </topic>
6      <topic id="ordem01">
7          <instanceOf>
8              <topicRef xlink:href="#ordem-de-compra"/>
9          </instanceOf>
10         <subjectIdentity>
11             <subjectIndicatorRef xlink:href="http://ordem-compra.com/01"/>
12         </subjectIdentity>
13         <baseName>
14             <baseNameString>Ordem 01</baseNameString>
15         </baseName>
16         <occurrence>
17             <instanceOf>
18                 <topicRef xlink:href="#data"/>
19             </instanceOf>
20             <resourceData>17/05/2005</resourceData>
21         </occurrence>
22         <occurrence>
23             <instanceOf>
24                 <topicRef xlink:href="#preco"/>
25             </instanceOf>
26             <scope>
27                 <topicRef xlink:href="#euro"/>
28             </scope>
29             <resourceData>1500</resourceData>
30         </occurrence>
31     </topic>

```

O tópico *ordem 01* (linha 6 a 31) – instância do tópico *ordem de compra* (linha 1 a 5) – possui um identificador de tema (linha 10 a 12), um nome (linha 13 a 15), uma ocorrência do tipo *data* (linha 16 a 21), uma ocorrência do tipo *preço* inserida no contexto *euro* (linha 22 a 30).

### 13.1.4 A relação *conter*

Quando se define uma ordem de compra, necessariamente presume-se houve a aquisição de produtos por algum cliente. Para tratar disso, inicialmente é definido o tipo desta relação, identificado pelo tópico *conter*, onde são definidos os nomes para os papéis de actuação neste tipo de associação. O tópico *conter*, abaixo definido, possui 3 nomes, onde o primeiro se refere ao papel de actuação *ordem de compra* (linha 5 a 10), o segundo ao *produto* (linha 11 a 16), e o terceiro à *quantidade* (linha 17 a 22).

```

1 | <topic id="conter">
2 |   <baseName>
3 |     <baseNameString>Conter</baseNameString>
4 |   </baseName>
5 |   <baseName>
6 |     <scope>
7 |       <topicRef xlink:href="#ordem-de-compra"/>
8 |     </scope>
9 |     <baseNameString>é composta pelos produtos</baseNameString>
10 |   </baseName>
11 |   <baseName>
12 |     <scope>
13 |       <topicRef xlink:href="#produto"/>
14 |     </scope>
15 |     <baseNameString>está incluído na ordem de compra</baseNameString>
16 |   </baseName>
17 |   <baseName>
18 |     <scope>
19 |       <topicRef xlink:href="#quantidade"/>
20 |     </scope>
21 |     <baseNameString>é o total de produtos da ordem de compra</baseNameString>
22 |   </baseName>
23 | </topic>

```

A associação entre *ordem de compra* e *produtos* conterà os identificadores dos tópicos envolvidos na relação, além da quantidade de produtos que foi adquirida. Abaixo, um exemplo deste tipo de associação é mostrado:

```

1 | <association>
2 |   <instanceOf>
3 |     <topicRef xlink:href="#conter"/>
4 |   </instanceOf>
5 |   <member>
6 |     <roleSpec>
7 |       <topicRef xlink:href="#ordem-de-compra"/>
8 |     </roleSpec>
9 |     <topicRef xlink:href="#ordem01"/>
10 |   </member>
11 |   <member>
12 |     <roleSpec>
13 |       <topicRef xlink:href="#produto"/>
14 |     </roleSpec>
15 |     <topicRef xlink:href="#dvd"/>
16 |   </member>
17 |   <member>
18 |     <roleSpec>
19 |       <topicRef xlink:href="#quantidade"/>
20 |     </roleSpec>
21 |     <topicRef xlink:href="#valor-2"/>

```

```

22 |         </member>
23 |     </association>

```

A associação acima indica que a *ordem 01* (linha 5 a 10) compreende a aquisição de dois produtos (linha 17 a 22), os quais são leitores de DVD (linha 11 a 16). Os tópicos *ordem 01* e *leitor de DVD* foram definidos nas subsecções anteriores. Pelo facto de que a quantidade é representada como parte da associação, o tópico com o número de unidades deve estar presente no topic map, o que se apresenta a seguir:

```

1 |     <topic id="valor-2">
2 |         <baseName>
3 |             <baseNameString>2</baseNameString>
4 |         </baseName>
5 |     </topic>

```

Desta forma, todas as associações do tipo *conter* possuem essa estruturação: um membro desempenhando o papel *ordem de compra*, um membro desempenhando o papel *produto* e um membro desempenhando o papel *quantidade*.

Quando mais de um tipo de produtos forem adquiridos numa mesma ordem de compra, será criada uma nova associação envolvendo o mesmo tópico do tipo *ordem de compra* com outro produto e a quantidade total. Por exemplo, a associação abaixo indica que a ordem de compra *ordem 01* também descreve a compra de 3 televisores.

```

1 |     <association>
2 |         <instanceOf>
3 |             <topicRef xlink:href="#conter"/>
4 |         </instanceOf>
5 |         <member>
6 |             <roleSpec>
7 |                 <topicRef xlink:href="#ordem-de-compra"/>
8 |             </roleSpec>
9 |             <topicRef xlink:href="#ordem01"/>
10 |        </member>
11 |        <member>
12 |            <roleSpec>
13 |                <topicRef xlink:href="#produto"/>
14 |            </roleSpec>
15 |            <topicRef xlink:href="#televisao"/>
16 |        </member>
17 |        <member>
18 |            <roleSpec>
19 |                <topicRef xlink:href="#quantidade"/>
20 |            </roleSpec>
21 |            <topicRef xlink:href="#valor-3"/>
22 |        </member>
23 |    </association>

```

A partir das duas associações descritas acima se entende que a ordem de compra *ordem 01* descreve a compra de dois leitores de DVD e de três televisores.

### 13.1.5 A relação *comprar*

As entidades *ordem de compra* e o *cliente* são relacionados através de associações do tipo *comprar*. Essa associação possui um membro desempenhando o papel *cliente* e no mínimo um membro desempenhando o papel *ordem de compra* (um cliente pode efectuar uma ou mais ordens). O tópico que define o tipo de associação *comprar* está abaixo descrito:

```

1 | <topic id="comprar">
2 |   <baseName>
3 |     <baseNameString>Comprar</baseNameString>
4 |   </baseName>
5 |   <baseName>
6 |     <scope>
7 |       <topicRef xlink:href="#cliente"/>
8 |     </scope>
9 |     <baseNameString>efectua a ordem de compra</baseNameString>
10 |   </baseName>
11 |   <baseName>
12 |     <scope>
13 |       <topicRef xlink:href="#ordem-de-compra"/>
14 |     </scope>
15 |     <baseNameString>foi realizada por</baseNameString>
16 |   </baseName>
17 | </topic>

```

Um exemplo de uma associação do tipo *comprar*, relacionando os tópicos *Giovani Librelotto* e *ordem 01*, é mostrado a seguir:

```

1 | <association>
2 |   <instanceOf>
3 |     <topicRef xlink:href="#comprar"/>
4 |   </instanceOf>
5 |   <member>
6 |     <roleSpec>
7 |       <topicRef xlink:href="#cliente"/>
8 |     </roleSpec>
9 |     <topicRef xlink:href="#grl"/>
10 |   </member>
11 |   <member>
12 |     <roleSpec>
13 |       <topicRef xlink:href="#ordem-de-compra"/>
14 |     </roleSpec>
15 |     <topicRef xlink:href="#ordem01"/>
16 |   </member>
17 | </association>

```

A partir das associações definidas até o momento se percebe que é construído um grafo envolvendo os tópicos *Giovani Librelotto*, *Ordem 01*, *televisão* e *leitor de dvd*.

Deste modo, além de indicar que o cliente *Giovani Librelotto* é o responsável pela ordem de compra *ordem 01*, percebe-se também que o referido cliente comprou *2 leitores de DVD* e *3 televisores*, pois estes produtos estão associados com a ordem de compra *ordem 01*.

## 13.2 Restrições sobre o topic map da *E-Sell*

Após a edição do topic map concreto, referido atrás, certas regras devem ser definidas para garantir a consistência dos dados contidos no topic map. O conjunto de restrições, descrito a seguir, é baseado nas regras impostas pelo caso de uso proposto na especificação de requisitos da TMCL (Nishikawa, Moore, and Bogachev, 2004):

### 1. Restrições sobre a entidade *cliente*:

- (a) Todos os tópicos do tipo *cliente*, devem possuir um nome, uma ocorrência para o endereço, um identificador de tema e, opcionalmente, uma ocorrência para o e-mail;
- (b) Os tópicos do tipo *cliente* devem também possuir uma ocorrência do tipo *fone* ou do tipo *fax*;
- (c) Ocorrências de endereço devem ser do tipo *endereço* e devem estar no contexto *Brasil*;
- (d) Ocorrências para e-mails devem ser do tipo *email* e conter o caracter “@”.
- (e) Tópicos do tipo *cliente* podem somente participar de associações do tipo *comprar*.
- (f) Identificador de tema para o *cliente-id* não pode conter quaisquer espaços ou caracteres ilegais;

### 2. Restrições sobre a entidade *produto*:

- (a) Todos os tópicos do tipo *produto* devem ter um identificador de tema, um nome, uma ocorrência para a quantidade de item em seu estoque, uma ocorrência para seu preço e um comentário;
- (b) As ocorrências do tipo *comentário* são opcionais;
- (c) O recurso de dados para o preço do produto deve ser do tipo *preço*, onde o valor é um *double* (restrição ao tipo de dados);
- (d) Tópicos do tipo *produto* podem somente participar de associações do tipo *conter*;
- (e) As ocorrências do tipo *preço* terão que estar no contexto *euro* ou no contexto *dolar*;

### 3. Restrições sobre a entidade *ordem de compra*:

- (a) Todos os tópicos do tipo *ordem de compra* devem ter um identificador de tema, um nome e ocorrências para a data de compra e o preço;
- (b) Todo tópico do tipo *ordem de compra* deve estar presente em apenas uma associação do tipo *conter*;
- (c) Todo tópico do tipo *ordem de compra* deve estar presente numa associação do tipo *comprar*.

4. Restrições sobre o relacionamento *comprar*:

- (a) Toda associação do tipo *comprar* deve envolver apenas 2 tópicos que irão desempenhar os papéis *cliente* e *ordem de compra* e que deverão ser do tipo *cliente* e *ordem de compra*, respectivamente;
- (b) O papel *cliente* pode ser executado por um só tópico;
- (c) O papel *ordem de compra* pode ser executado por um ou mais tópicos.

5. Restrições sobre o relacionamento *conter*:

- (a) Toda associação do tipo *conter* deve envolver 3 tópicos que desempenham os papéis *ordem de compra*, *produto* e *quantidade*;
- (b) Toda associação do tipo *conter* deve ter um tópico membro executando o papel de actuação *quantidade*.
- (c) Toda associação do tipo *conter* deve ter um tópico membro do tipo *ordem de compra* executando o papel de actuação *ordem de compra*;
- (d) Toda associação do tipo *conter* deve ter um tópico membro do tipo *produto* executando o papel de actuação *produto*;

6. Restrições contextuais:

- (a) O tópico *comprar* somente pode ser um tipo de associação;
- (b) O tópico *conter* somente pode ser um tipo de associação;
- (c) O tópico *cliente* pode ser um tipo de tópico ou um papel em associação;
- (d) O tópico *ordem de compra* pode ser um tipo de tópico ou papel em associação;
- (e) O tópico *produto* pode ser um tipo de tópico ou um papel em associação;

### 13.3 Restrições XTche sobre o topic map da *E-Sell*

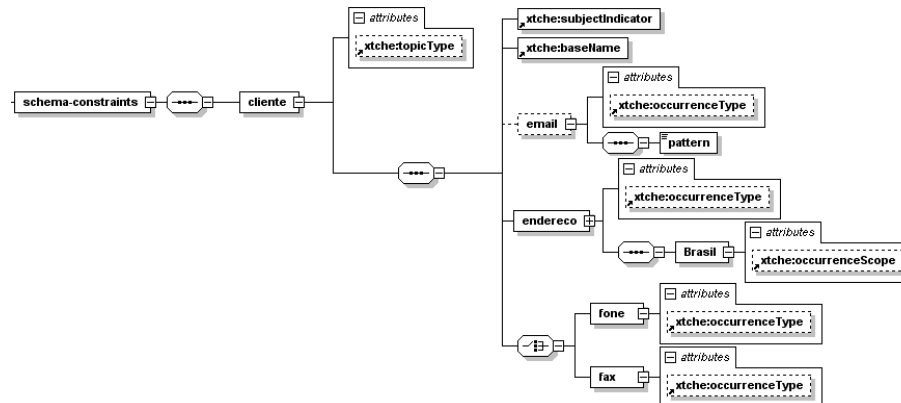
A linguagem XTche permite a especificação de todas as restrições apresentadas no âmbito do caso de estudo da *E-Sell*. As próximas subsecções tratam cada caso citado na secção anterior, de forma particular. Note-se que os 5 primeiros grupos de restrição são de estrutura; só o sexto que engloba restrições de contexto, conforme o nome indica.

#### 13.3.1 Restrições sobre a entidade *cliente*

Dos três tipos de tópicos encontrados no domínio do caso de estudo da *E-Sell*, o tipo *cliente* é o que apresenta a maior riqueza semântica, conforme se vê na restrição XTche da Figura 13.2, a qual abrange as restrições 1(a), 1(c) e 1(d) da Secção 13.2.

O elemento *cliente* é definido como um sub-elemento de **schema-constraints** porque ele formaliza a especificação de uma restrição de esquema.



Figura 13.2: Especificação XTche do tipo de tópico *Cliente*

O elemento *cliente* possui um atributo `xtche:topicType` – indicando que ele define um tipo de tópico – e é composto então por uma sequência de cinco sub-elementos.

O primeiro sub-elemento de *cliente* é `xtche:subjectIndicator`, o qual indica que todo tópico deste tipo possuirá um indicador de tema, independente de qual for o seu conteúdo.

O segundo corresponde ao nome dos tópicos deste tipo. Usa-se a declaração `xtche:baseName` para definir que todo tópico deste tipo terá um nome sem tipo e sem contexto, pois a restrição definida na secção anterior não prevê qualquer uma dessas situações.

O terceiro sub-elemento chama-se *email*, que possui o atributo `xtche:occurrenceType`. Portanto, *email* é um tipo de ocorrência dos tópicos do tipo *cliente*. De acordo a regra a ser aplicada, a ocorrência do tipo *email* é opcional; por isso, define-se que o número de ocorrências mínimas deste elemento é 0 (zero).

Completando a especificação da ocorrência *email*, o sub-elemento *pattern* define que toda ocorrência deste tipo deve conter o caracter “@”. Para isto, especifica-se uma restrição contendo uma expressão XSL (Holman, 2001), conforme apresentado abaixo<sup>2</sup>:

```

1 <xs:element name="pattern">
2   <xs:simpleType>
3     <xs:restriction base="xs:string">
4       <xs:pattern value="contains('@')"/>
5     </xs:restriction>
6   </xs:simpleType>
7 </xs:element>

```

O elemento *endereco*, com o atributo `xtche:occurrenceType`, também define um tipo de ocorrência. Além disso, toda ocorrência deste tipo necessariamente estará dentro do contexto “*Brasil*”, pois seu sub-elemento está qualificado com o atributo `xtche:occurrenceScope`.

A satisfação da restrição 1(b) é garantida pelo elemento *choice* encontrado na Figura 13.2. Ele possui dois sub-elementos (*fone* e *fax*) e sua semântica indica que todo tópico do tipo

<sup>2</sup>Tal especificação XML Schema não tem representação gráfica na imagem gerada pelo XML Spy®, por isto é apresentado textualmente.

*cliente* deve ter uma opção ou outra; ou seja, ou o cliente tem um número de fax, ou um número de telefone.

A restrição 1(e) da Secção 13.2 é descrita na Subsecção 13.3.4, pois envolve a associação *Comprar*.

Por sua vez, a restrição 1(f) é resolvida pela própria norma ISO 13250, pois os indicadores de tema não podem conter caracteres ilegais ou espaços. Portanto, não é um tipo de restrição com a qual XTche ou outra linguagem de restrição se deva preocupar.

### 13.3.2 Restrições sobre a entidade *produto*

Tomando como base a especificação do tipo de tópico *cliente*, o tipo de tópico *produto* pode ser facilmente definido, pois as restrições de esquema são similares às aquelas apresentadas anteriormente.

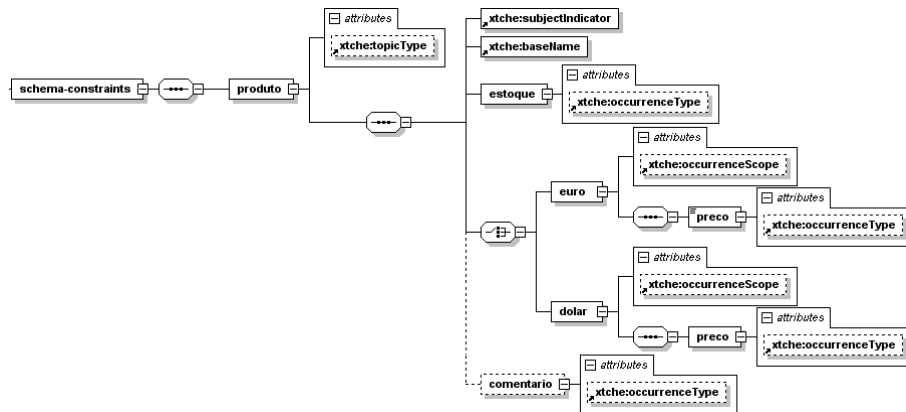


Figura 13.3: Especificação XTche do tipo de tópico *produto*

Para definir a restrição 2(a), o elemento *produto* (com o atributo `xtche:topicType`) possui 5 sub-elementos, de acordo com a Figura 13.3: o primeiro define que todo tópico deste tipo terá um indicador de tema, enquanto que o segundo diz que um nome também é obrigatório aparecer, o terceiro define a ocorrência do tipo *estoque*, o quarto a ocorrência referente ao *preço* e o quinto a ocorrência do tipo *comentário*.

O último sub-elemento determina que todo produto pode ter comentários, satisfazendo a restrição 2(b). O elemento *comentário* é definido como opcional, usando o atributo `minOccurs` com o valor 0, como visto a seguir:

```
1 | <xs:element name="comentario" minOccurs="0">
```

O quarto elemento é um *choice* com dois sub-elementos: *euro* e *dólar*, onde ambos possuem o atributo `xtche:occurrenceScope`. O *choice* representa uma escolha: todo tópico do tipo *produto* terá um preço em euros (contexto *euro*) ou um preço em dólares (contexto *dólar*). Isso satisfaz a restrição 2(e).

Os elementos chamados *preço* satisfazem a restrição 2(c), pois além deles definirem um tipo de ocorrência, com atributo `xtche:occurrenceType` – para completar a restrição 2(a) – o seu conteúdo define que o tipo de dados que ele vai obedecer é *xs:double*, um dos tipos de dados de XML Schema, conforme definido abaixo:

```

1 <xs:element name="preço">
2   <xs:complexType>
3     <xs:simpleContent>
4       <xs:extension base="xs:double">
5         <xs:attribute ref="xtche:occurrenceType"/>
6       </xs:extension>
7     </xs:simpleContent>
8   </xs:complexType>
9 </xs:element>

```

Desta forma, o conteúdo da ocorrência do tipo *preço* fica limitada a valores definidos pelo tipo de dado *double*.

### 13.3.3 Restrições sobre a entidade *ordem de compra*

Assim como as restrições para o tipo de tópico *produto*, as regras XTche para o tipo de tópico *ordem de compra* seguem o conjunto de regras definidos para o tipo de tópico *cliente*.

A restrição 3(a) da Secção 13.2 define que todo tópico deste tipo possuem um identificador de tema especificado no elemento `xtche:subjectIndicator` apresentado na Figura 13.4, o qual é um sub-elemento de *ordem de compra*. Além disso, é definido um sub-elemento para o nome e outros 2 que determinam os tipos de ocorrência deste tipo de tópico: *data* da compra e *preço*.

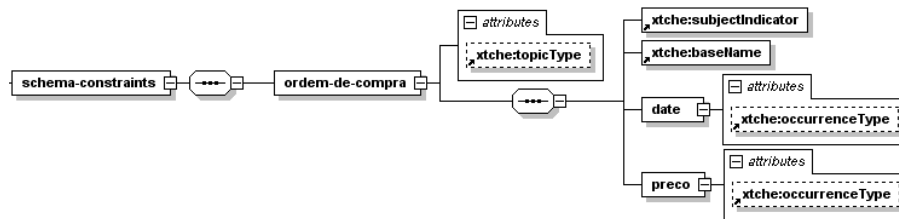


Figura 13.4: Especificação XTche do tipo de associação *ordem de compra*

A restrição 3(b) está definida na Subsecção 13.3.4, pois toda associação do tipo *comprar* deve ter um papel do tipo *ordem de compra*. Por sua vez, a Subsecção 13.3.5 satisfaz a restrição 3(c), pois apresenta que tópicos do tipo *ordem de compra* podem desempenhar papel de actuação em associações do tipo *conter*.

### 13.3.4 Restrições sobre o relacionamento *comprar*

A definição do conjunto de regras sobre a associação *comprar* baseia-se nas restrições 4(a), 4(b) e 4(c) propostas na Secção 13.2.

A Figura 13.5 mostra esta restrição. O elemento *comprar* define o tipo de associação (`xtche:associationType`) que é composto pelos papéis de actuação *cliente* e *ordem* (`xtche:associationRole`), conforme diz a restrição 4(a).

Para satisfazer a restrição 4(b) define-se, como sub-elemento do papel *cliente*, o elemento *cliente* com os atributos `xtche:topicType` e `xtche:associationPlayer-Exclusive`. O primeiro atributo indica que o papel *cliente* só será desempenhado por um membro instância do tipo de tópicos *cliente*. O segundo atributo resolve a condição imposta pela restrição 1(e), pois o sufixo **Exclusive** define que um tópicos do tipo *cliente* só pode ser membro do tipo de associação corrente, ou seja, só pode ser envolvido em associações do tipo *comprar*.

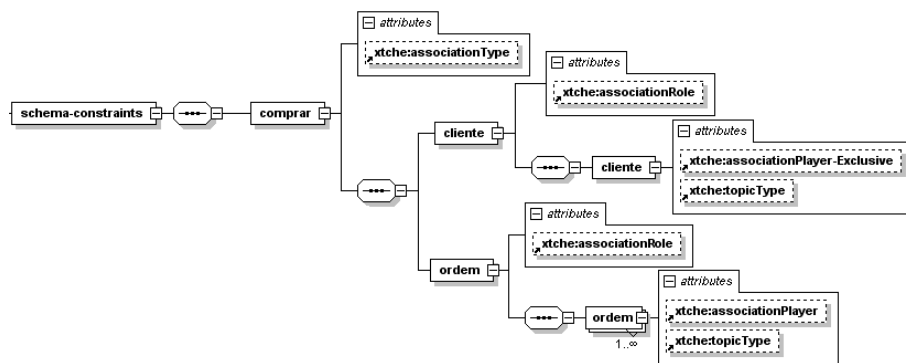


Figura 13.5: Especificação XTche do tipo de associação *comprar*

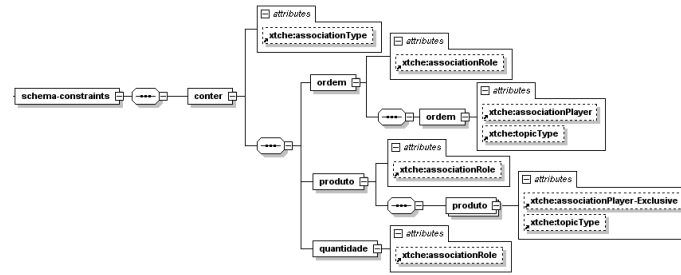
A segunda parte da regra apresentada na Figura 13.5 completa a restrição 4(c) da Secção 13.2. O elemento *ordem* (que possui a cardinalidade entre o valor 1 e ilimitado) e os atributos `xtche:associationPlayer` e `xtche:topicType` determinam que o papel de actuação *ordem* será desempenhado por um ou mais tópicos do tipo *ordem*.

### 13.3.5 Restrições sobre o relacionamento *conter*

A definição das restrições sobre o relacionamento *conter* são similares às apresentadas na subsecção anterior. A restrição 5(a) da Secção 13.2 é especificada pelos elementos *ordem* e *produto* com o atributo `xtche:associationRole`, conforme visto na Figura 13.6. O elemento *quantidade* satisfaz a restrição 5(b).

A restrição 5(c) especifica-se com o elemento *ordem* – sub-elemento do elemento que define o papel de actuação *ordem* – com os atributos `xtche:topicType` e `xtche:associationPlayer`.

A restrição 5(d) especifica-se com o elemento *produto* – sub-elemento do elemento que define o papel de actuação *produto* – com os atributos `xtche:topicType` e `xtche:associationPlayer-Exclusive`. Este último atributo ainda especifica a restrição 2(d), pois determina que todo tópicos do tipo *produto* só pode ser membro de associações do tipo *conter*.

Figura 13.6: Especificação XTche do tipo de associação *Conter*

### 13.3.6 Restrições Contextuais

A especificação das restrições contextuais é simples e rápida de ser realizada. A Figura 13.7 apresenta graficamente as especificações do item 6 da Secção 13.2, de acordo com a linguagem XTche.

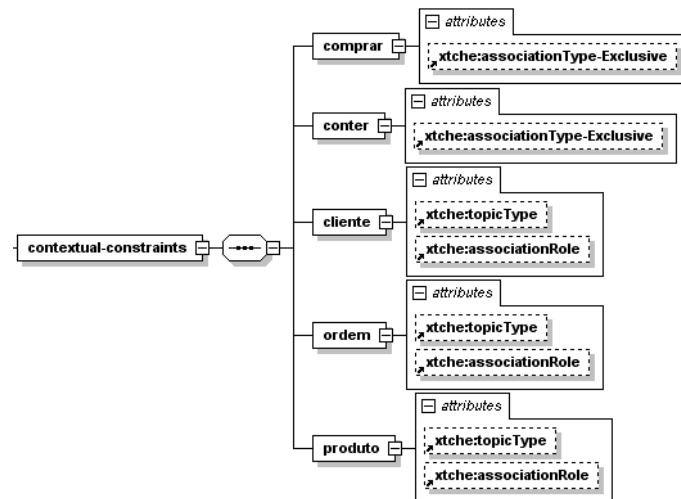


Figura 13.7: Especificação em XTche das restrições contextuais

Os cinco elementos encontrados no contexto de `contextual-constraints` possuem dois tipos de definições:

**Tipos de associação:** os elementos *comprar* (restrição 6(a) da Secção 13.2) e *conter* (restrição 6(b)) são especificados como tipos exclusivos de associação. Portanto, eles não podem ser encontrados no domínio deste topic map desempenhando nenhuma outra função, a não ser tipo de associação. Em XTche, usa-se o atributo `xtche:associationType-Exclusive` para definir este tipo de restrição;

**Tipos de tópico e papel de actuação em associação:** os elementos *cliente*, (restrição 6(c)), *ordem de compra* (restrição 6(d)) e *produto* (restrição 6(e)) são definidos contendo os atributos `xtche:topicType` e `xtche:associationRole`, o que significa que esses

tópicos podem apenas ser utilizados para a definição de tipos de tópicos, ou para papéis de actuação em associação. Não se usa aqui o sufixo `exclusive` porque cada elemento pode assumir duas ou mais formas no topic map (neste caso, cada tópico pode ser usado como tipo de tópico ou papel de actuação em associação).

Além destes elementos, poderiam ser especificados outros como *email*, *endereço*, *fone*, *fax*, *preço*, etc, os quais são tipos exclusivos de ocorrência. Para sua especificação, bastaria a criação de um novo sub-elemento com cada nome, com o atributo `xtche:occurrenceType-Exclusive`.

### 13.4 Processamento da especificação XTche e validação do topic map da *E-Sell*

Para a codificação da especificação XTche, apresentada na secção anterior, construiu-se um documento de acordo com a sintaxe XML Schema com 216 linhas (Figura 13.8). Para o teste desta especificação, criou-se um topic map com 142112 linhas, contendo 855 tópicos e 3141 associações.

A especificação XTche está num ficheiro chamado `e-sell-corp-xtche.xsd`, enquanto o topic map referente ao domínio da *E-Sell* está em outro ficheiro com nome `e-sell-corp.xtm`.

O processador de XTche invoca-se na linha de comando, passando os 2 ficheiros anteriores, XTche e XTM, como parâmetros, conforme apresentado abaixo:

```

1 C:\XTche>xtche e-sell-corp-xtche.xsd e-sell-corp.xtm
2 -----
3           XTche - Topic Maps Validator
4           -----
5           Errors: 0. See "error.xml" for details.
6 -----

```

A primeira fase deste processamento é responsável pela criação, a partir da especificação XTche, de um TM-Validator, uma folha de estilos XSL com 2516 linhas; o TM-Validator é então invocado de imediato e processa o topic map fornecido como parâmetro; o TM-Validator informa o utilizador o total de número de erros encontrados, armazenando-os no ficheiro chamado `error.xml`.

Nesse processamento não foi detectado qualquer erro no topic map; ou seja, o topic map segue com rigidez todas as regras definidas na especificação XTche referente. Para simular um erro, foi alterada uma ocorrência do tipo *endereço* do tópico *Giovani Librelotto* (identificado por *grl*). A ocorrência, que antes estava no contexto *Brasil*, agora está no contexto *Portugal*. Após o processamento deste novo topic map, agora armazenado num ficheiro chamado `e-sell-corp-wrong.xtm`, o resultado é o abaixo mostrado:

```

1 C:\XTche>xtche e-sell-corp-xtche.xsd e-sell-corp-wrong.xtm
2 -----
3           XTche - Topic Maps Validator
4           -----
5           Errors: 1. See "error.xml" for details.
6 -----

```

Agora, um erro foi detectado e guardado no ficheiro de *log* chamado `error.xml`. O seu conteúdo apresenta os erros encontrados na avaliação (no caso, apenas um):

```

1 <?xml version="1.0" encoding="utf-8" standalone="yes"?>
2 <doc-status xmlns="http://www.di.uminho.pt/~gepl/xtche/namespace">
3   <err-message number="1">Constraint: 4.1.2 - U2 - (d) : Topic #grl must have an occurrence
4     of type #endereco in scope #Brasil.</err-message>
5 </doc-status>

```

A nomenclatura “*Constraint: 4.1.2 - U2 - (d)*” é baseada na especificação de requisitos associada à linguagem TMCL (Nishikawa, Moore, and Bogachev, 2004). Esse documento possui, na Secção 4 (*TMCL Use Cases*), os casos particulares de aplicação da linguagem TMCL. A Subsecção 4.1.2, item U2, parágrafo *d*, diz:

4.1.2 - U2 - (d): Topic of type T must have an occurrence in scope S.

Desta forma indica-se o tipo de restrição, definido nos requisitos de TMCL, que foi desobedecido pelo topic map em avaliação.

O ficheiro de *log* abaixo refere-se a uma avaliação de um topic map com uma quantidade significativa de divergências perante a especificação XTche apresentada neste capítulo. O topic map avaliado possui um total de 42 erros de 13 tipos distintos.

```

1 <?xml version="1.0" encoding="utf-8" standalone="yes"?>
2 <doc-status>
3   <err-message number="4">Constraint: 4.1.1 - U8 - (a) : Association of type #contem must have
4     the role #produto being played by a topic of type #produto.</err-message>
5   <err-message number="1">Constraint: 4.1.2 - U2 - (d2) : Topic #grl must have an occurrence of
6     type #endereco in scope #Brasil.</err-message>
7   <err-message number="1">Constraint: 4.1.2 - U2 - (d2) : Topic #marco must have an occurrence of
8     type #endereco in scope #Brasil.</err-message>
9   <err-message number="5">Constraint: 4.1.1 - U8 - (a) : Association of type #conter must have
10    the role #ordem being played by a topic of types #ordem.</err-message>
11   <err-message number="1">Constraint: 4.1.3 - U4 - (a) : Occurrence of type #email needs to
12    contain the word '@'.</err-message>
13   <err-message number="6">Constraint: 4.1.1 - U4 - (a2) : Topic #ordem can only be used as an
14    association role topic.</err-message>
15   <err-message number="5">Constraint: 4.1.2 - U4 - (a1) : Topic #cliente can be used for typing
16    other topics and nothing else. Error in an association role.</err-message>
17   <err-message number="1">Constraint: 4.1.2 - U4 - (a1) : Topic #cliente can be used for typing
18    other topics and nothing else. Error in Topic #comprar.</err-message>
19   <err-message number="1">Constraint: 4.1.2 - U4 - (b2) : Topic #preco can be used for typing
20    other occurrences and nothing else. Error in Topic #dvd.</err-message>
21   <err-message number="2">Constraint: 4.1.1 - U2 - (a) : Association with association type
22    #comprar must be in scope #portugues.</err-message>
23   <err-message number="5">Constraint: 4.1.1 - U4 - (a1) : Topic #cliente must be declared to be
24    a association role.</err-message>
25   <err-message number="9">Constraint: 4.1.1 - U4 - (a1) : Topic #ordem must be declared to be a
26    association role.</err-message>
27   <err-message number="1">Constraint: 4.1.2 - U4 - (b3) : Topic #produtos can not be used for
28    typing occurrences. Error in Topic #televisao.</err-message>
29 </doc-status>

```

O atributo `number` indica quantas vezes o ocorre um mesmo erro, em toda a avaliação do topic map. Por exemplo, as linhas 3 e 4 dizem que o erro *4.1.1 - U8 - (a)* é encontrado

4 vezes durante o processo de avaliação. Da mesma forma, cada erro tem este atributo indicando quantas vezes ele foi detectado.

### 13.5 Sumário do caso de estudo da *E-Sell Corporation*

Este capítulo apresentou a abordagem utilizada por XTche para resolver o caso de estudo encontrado na especificação de requisitos associada à definição da TMCL (Nishikawa, Moore, and Bogachev, 2004), proposto pela ISO. Portanto, o intuito deste capítulo foi comprovar que a linguagem XTche permite a definição das restrições exigidas pela comunidade académica de Topic Maps.

O caso de estudo em questão foi estendido, em relação ao proposto pela ISO, com a inclusão de outras cláusulas, como por exemplo: condições opcionais (restrição 2(b) da Secção 13.2) e alternativas (restrição 5(b)). Essas alterações mostram que a linguagem XTche tem um poder de expressividade maior que o exigido pelo caso de estudo.

Para os testes da especificação XTche referente ao domínio da *E-Sell*, foram construídos dois topic map: o primeiro deles obedecia correctamente à estrutura definida na Secção 13.1; o segundo, foi sendo alterado manualmente de forma a testar todas as condições verificadas pelas regras XTche.

No primeiro caso, o TM-Validator – obtido a partir da especificação XTche – não indicou qualquer erro, o que de facto foi verificado de forma exaustiva pelo editor do topic map. Por sua vez, o segundo caso indicou os erros conforme eles eram inseridos no topic map: sejam eles de ordem estrutural ou contextual.

Após uma bateria de testes – seja com topic maps que obedeciam às regras definidas, ou com topic maps inválidos – chegou-se à conclusão que a linguagem XTche realmente supre as necessidades impostas pelos requisitos de TMCL.

A visualização gráfica do conjunto completo de restrições a serem impostas ao topic map da *E-Sell* encontra-se na Figura 13.8.



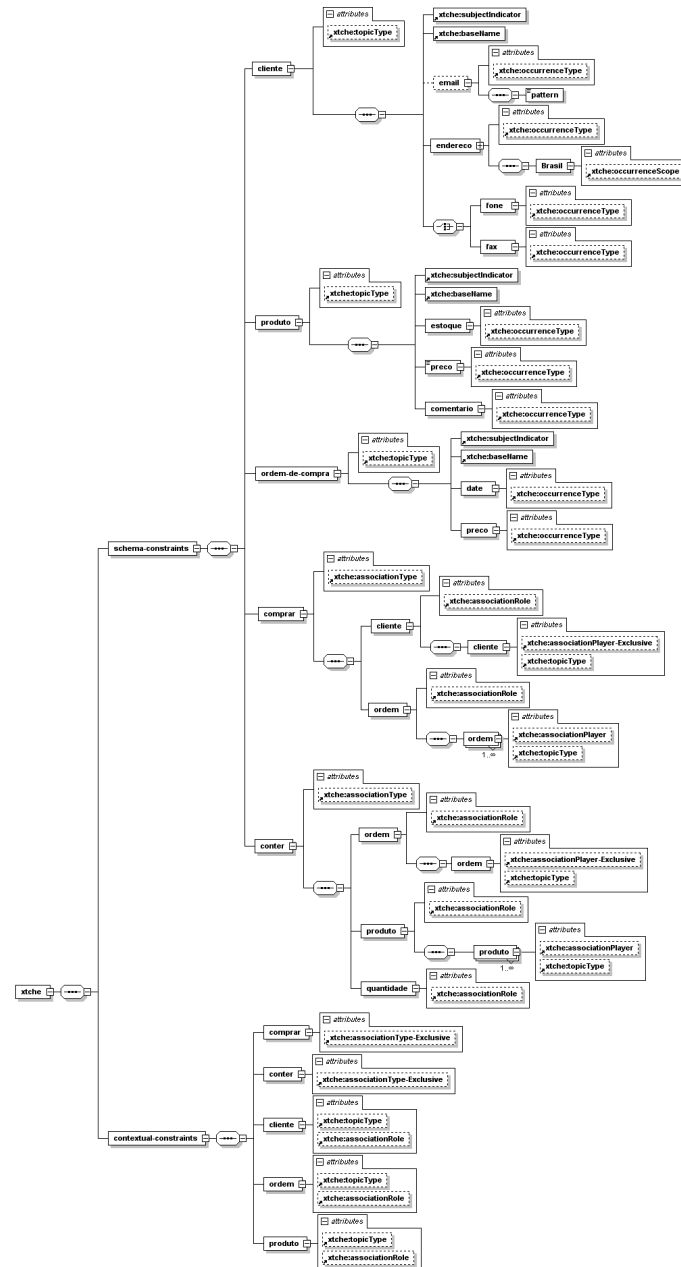


Figura 13.8: Especificação XTche completa para o caso de estudo de *E-Sell*



## Capítulo 14

# O Metamorphosis aplicado ao Museu da Emigração

*A um bochincho - certa feita,  
Fui chegando - de curioso,  
Que o vicio - é que nem sarnoso,  
nunca pára - nem se ajeita.  
Baile de gente direita  
Vi, de pronto, que não era,  
Na noite de primavera  
Gaguejava a voz dum tango  
E eu sou louco por fandango Que nem pinto por quirela.”*

Jayme Caetano Braun

Devido ao facto de as pessoas viverem em constante mudança, o ritmo das suas vidas as faz incorrer em viagens (por puro lazer, por necessidade ou por trabalho). Num passado não muito longínquo, era frequente encontrar em almanaques locais, histórias sobre a vida de determinados indivíduos ou sobre eventos em que os mesmos participavam. Também era usual fazer-se um diário de viagem quando se realizava uma jornada por vários locais (países, cidades, etc).

O *Museu da Emigração*<sup>1</sup> (ME) foi criado para reunir e disponibilizar à comunidade informações diversas acerca dos emigrantes em geral e, em particular, daqueles que emigraram para o Brasil.

Particularmente importante nesse sentido é que o ME é um museu virtual. Para tal, a informação a ser exibida no ME é recolhida a partir de diversas fontes de dados, tais como registos de passaportes, almanaques, jornais, diários de viagem e certidões de nascimento. Esta lista de possíveis fontes de dados não está limitada, pois novas fontes (recheadas de informação relevante para o ME) podem sempre aparecer.

---

<sup>1</sup>Museu da Emigração – <http://www.museu-emigrantes.webside.pt>

Combinando várias fontes – como por exemplo, as fontes que contém dados sobre os almanaques, diários de viagem e passaportes – pode-se disponibilizar uma quantidade significativa de informação sobre um determinado indivíduo, reconstruindo-se assim o seu percurso de vida que irá contextualizar e enquadrar o seu processo de emigração.

Os *almanaques* são publicações, normalmente periódicas e nacionais (embora por vezes possam ter um cariz mais local), com calendário, informações científicas, tabelas, informações úteis sobre o quotidiano e alguns textos mais amenos como contos, notas biográficas, descrição de eventos sociais, entretenimento, etc. *Diários de viagem* são relatos sucintos de suas jornadas, ou seja, pequenas anotações que certas pessoas faziam, contendo informações como a data, o local e uma descrição textual de sítios visitados (monumentos ou paisagens) ou episódios de viagem. As informações sobre os *passaportes* indicam para onde cada emigrante se deslocou, assim como quem eram seus acompanhantes, o motivo da viagem, etc.

Pensando agora em termos do Sistema de Informação digital (SIME) que irá suportar o Museu Virtual da Emigração, é importante observar que, pela natureza dos documentos aqui envolvidos e por razões práticas e factuais, as fontes de informação estão armazenadas em suportes heterogéneos: algumas são guardadas em bases de dados relacionais, enquanto outras estão na forma de textos anotados em XML.

Neste capítulo, pretende-se apresentar o uso do Metamorphosis para a integração de *fontes nominativas históricas heterogéneas*, isto é, armazenadas de formas diversas em suportes diferentes. O objectivo é propiciar à comunidade em geral a extracção do conhecimento a partir das fontes de dados do Museu da Emigração, através da navegação conceptual sobre a informação contida nas seguintes fontes heterogéneas:

- uma base de dados relacional contendo o conjunto de passaportes catalogados;
- documentos anotados em XML sobre diários de viagem dos emigrantes;
- documentos anotados em XML sobre as notas biográficas (histórias de vida) recolhidas nos almanaques.

O Metamorphosis realiza então três passos para obter a interoperabilidade entre tais fontes de dados:

1. o Oveia extrai a informação dos recursos, criando um topic map contendo o conhecimento do universo de discurso especificado em XS4TM;
2. o topic map é validado pelo XTche;
3. a navegação conceptual sobre este topic map é fornecida pelo Ulisses.

Assim, pode-se afirmar que este caso de estudo é ideal para fechar a 3ª Parte desta dissertação, uma vez que ilustra o uso integrado e integral de todos os componentes do sistema Metamorphosis proposto nesta tese (Capítulo 6).

Este capítulo encontra-se dividido em 6 secções. A Secção 14.1 descreve as fontes de dados que fazem parte do SIME, apresentando sua estrutura e conteúdo. A Secção 14.2 define os 3 passos a serem seguidos para a obtenção de uma navegação sobre o domínio do Museu da Emigração: o primeiro passo, apresentado na Secção 14.3, descreve como o Oveia propicia a geração de um topic map a partir de recursos heterogéneos; a Secção 14.4 especifica um conjunto de regras através das quais XTche valida o topic map gerado; por fim, a Secção 14.5 disponibiliza a visualização do domínio do ME através de páginas HTML geradas pelo Ulisses. A conclusão do caso de estudo em questão se apresenta na Secção 14.6.

## 14.1 Sistema de Informação do Museu da Emigração

Normalmente a informação relevante que se pretende extrair de qualquer sistema está armazenada em formatos distintos. Este foi o caso encontrado no âmbito do SIME.

Os dados que alimentam esse sistema estão armazenados em documentos anotados em XML (para as fontes de dados referentes aos *Almanaques* e aos *Diários de Viagem*), de onde pode-se extrair as histórias da vida e de viagens referentes aos emigrantes. Combinando estes documentos XML com uma base de dados que contém a informação relativa aos seus passaportes, fica-se com um conjunto de informações sobre o percurso de vida e algumas histórias com relevo de emigrantes.

As próximas subsecções descrevem a estrutura e exemplificam o conteúdo de cada fonte de dados pertencente ao SIME.

### 14.1.1 Notas Biográficas contidas em Almanaque

Ao longo do tempo, os *Almanaques* foram trazidos a público – com uma periodicidade semanal ou mensal – por algumas localidades ou grupo de pessoas. Com isso, uma enorme quantidade de almanaques foram reunidos pelo Museu da Emigração, em seu formato original (em papel). A capa de um almanaque é visualizada na Figura 14.1.

Contudo, os almanaques não se encontravam escritos de uma forma facilmente manipulável (de um ponto de vista informático). De um modo geral, as histórias dos almanaques não possuem um conteúdo semelhante entre si, podendo abranger um grande leque de descrições sociais, psicológicas, físicas, além de conter informação ambígua e parcial.

Um exemplo de uma história de vida retirada de um Almanaque de Fafe é encontrado na Figura 14.2, onde se percebe que as informações acerca do emigrante “José Pereira Leite” são apresentadas sem uma estruturação definida. Contudo, pode-se obter várias informações sobre este emigrante, tais como:

- uma definição de seu perfil: amigo, sincero, dedicado, ...;
- sua naturalidade: Chacim, Cabeceiras de Basto;



Figura 14.1: Exemplo da capa de um Almanaque.

- para onde emigrou: Brazil;
- seus bens: boa casa de negócio;
- seu colega de trabalho: Joaquim de Macedo Ferraz e Souza.

Para seu armazenamento digital, definiu-se então uma estrutura XML que permita a extração e catalogação das informações possíveis de ser encontradas nas Notas Biográficas destes almanaques. O respectivo esquema XML gerado pelo XMLSpy® 2005, apresentado na Figura 14.3, define a estrutura lógica de um almanaque anotado em XML. Abaixo encontra-se a anotação em XML do exemplo de história de vida do emigrante “*José Pereira Leite*” apresentado na Figura 14.2.

```

1 <Notas_Biograficas>
2   <fonte>Almanaque de Fafe</fonte>
3   <titulo>José Pereira Leite</titulo>
4   <tipoMemoria valor="outro"/>
5   <MEMORIA>
6     <PESSOA ID="ID000006">
7       <nome>José Pereira Leite</nome>
8       <profissao>Comércio</profissao>
9     </PESSOA>
10    <HISTORIA_A>
11      <notasAdicionais>
12        <perfil valorPerfil="psicologico">Sincero e Dedicado</perfil>
13      </notasAdicionais>
14      <localNasc>Natural de Chacim, Cabeceiras de Basto</localNasc>
15      <paisEmigrou>Brasil</paisEmigrou>
16      <notasAdicionais>

```



Figura 14.2: Um exemplo de uma história de vida retirada do mesmo Almanaque.

```

17         <perfil valorPerfil="profissional">Graças à sua força de vontade e
18 actividade, possui uma boa casa de negócio</perfil>
19 <relSociais>Tem ao seu serviço o Sr. Joaquim de Macedo Ferraz e Souza,
20 conhecidíssimo na vila de Fafe</relSociais>
21 <perfil valorPerfil="psicologico">Modesto e Bom</perfil>
22 </notasAdicionais>
23 <docAnexo>foto6</docAnexo>
24 </HISTORIA_A>
25 </MEMORIA>
26 </Notas_Biograficas>

```

Esta história anotada em XML possui a mesma informação acerca desse emigrante que a página de um almanaque apresentada na Figura 14.2. Porém, a informação está estruturada e explicitamente marcada (para permitir uma interpretação objectiva, não ambígua), tornando viável o seu arquivo electrónico e o processamento deste documento por um parser XML. As notas biográficas retiradas dos Almanques de Fafe no âmbito do ME estão juntamente armazenadas em ficheiros de texto que seguem o formato acima exemplificado.

### 14.1.2 Diários de Viagem

Um *Diário de Viagem* é um documento onde uma pessoa escreve possíveis notas de embarque, repouso e o que fez ou não em certos locais. Estas notas são acompanhadas da data do respectivo apontamento. Resumindo, é um diário que apenas se singe a uma dada viagem efectuada por uma pessoa.

Os diários de viagem são muito genéricos, sucintos e podem ocorrer grandes intervalos de tempo em que não existe informação (pode não existir uma descrição contínua, metódica e temporal do relato da viagem). Um exemplo de um excerto de um diário de viagem encontra-se na Figura 14.4.

Assim como no caso dos almanques, foi definido um esquema genérico o suficiente, tornando possível a catalogação de toda a informação contida num diário. A Figura 14.5

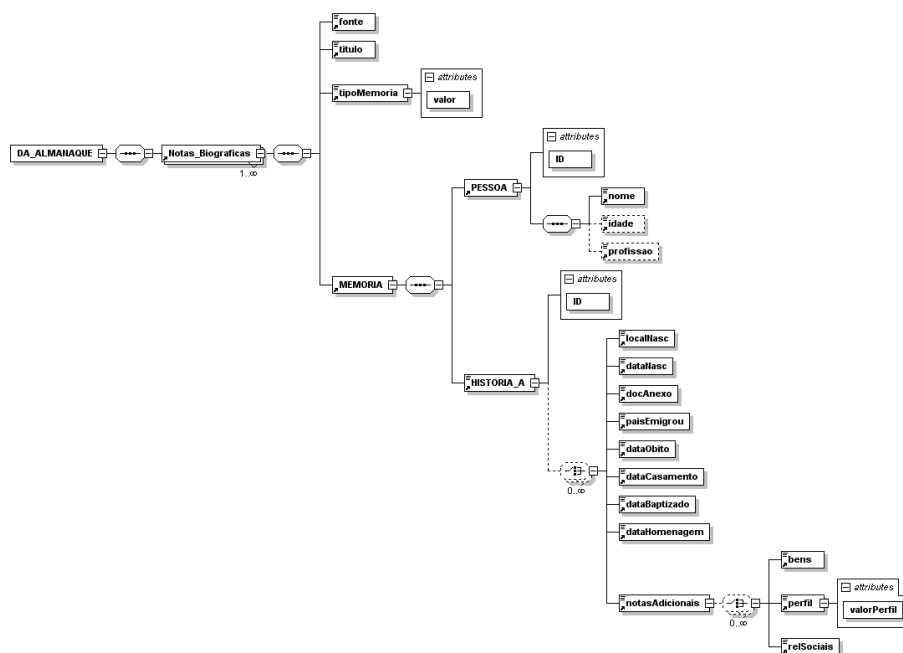


Figura 14.3: Esquema dos Almanagues.

apresenta o esquema projectado.

Este esquema define a estrutura lógica, em anotação XML, para um diário de viagem, o qual é exemplificado pelo documento abaixo:

```

1 <Notas_Bio>
2   <PESSOA ID="ID000006">
3     <nome>José Pereira Leite</nome>
4     <idade>25</idade>
5   </PESSOA>
6   <HISTORIA_V ID="HIST0100">
7     <nota data="1905-01-07">Início da viagem à Europa: Amanhã.</nota>
8     <nota data="1905-01-08">Sahida hoje do Rio, para a Bahia, as 4 horas da tarde
9     pelo Clíjide.</nota>
10    <nota data="1905-01-09">...</nota>
11    <nota data="1905-01-10">...</nota>
12  </HISTORIA_V>
13 </Notas_Bio>

```

Em termos do SIME, todos os diários de viagem recolhidas e pertencentes ao âmbito do ME encontram-se anotados em XML, tal qual o exemplo acima.

### 14.1.3 Registos de Passaportes

Os passaportes contém a informação referente às viagens efectuadas por um emigrante, informando qual foi o destino da viagem (podendo ser múltiplos destinos) e se ele foi acompanhado por alguma outra pessoa (podendo ser múltiplos acompanhantes).



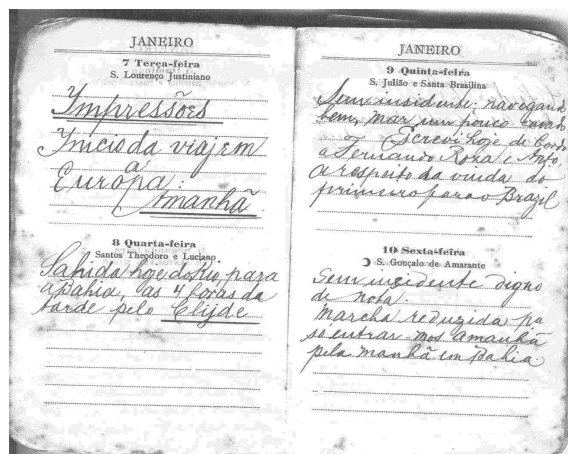


Figura 14.4: Excerto de um Diário de Viagem.

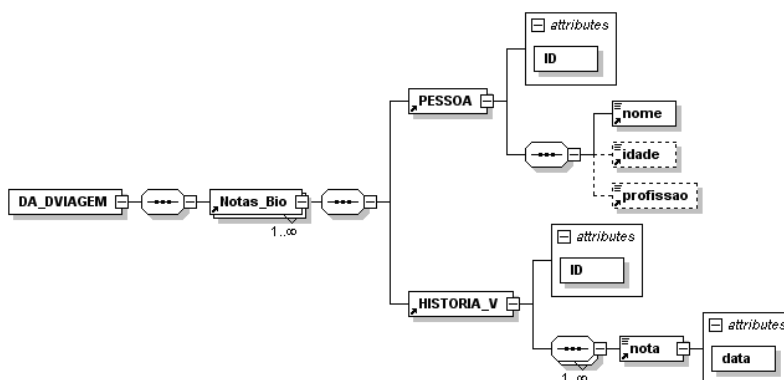


Figura 14.5: Esquema de um Diário de Viagem.

A informação referente aos passaportes está estruturada seguindo o modelo relacional em quatro tabelas: *Passaportes* (contém a informação base dos passaportes), *Viagem* (contém as informações referentes às viagens efectuadas por determinado emigrante), *Acompanhantes* (contém todas as informações sobre os acompanhantes dos emigrantes em cada viagem) e *Pessoa* (contém os dados pessoais de cada emigrante). A Figura 14.6 apresenta estas quatro tabelas e os relacionamentos entre elas, no diagrama entidade-relacionamento dessa base de dados.

Os relacionamentos da Figura 14.6 indicam que uma pessoa pode possuir vários passaportes, assim como ter vários acompanhantes associados a ela. Por sua vez, cada passaporte poderá referenciar vários destinos.

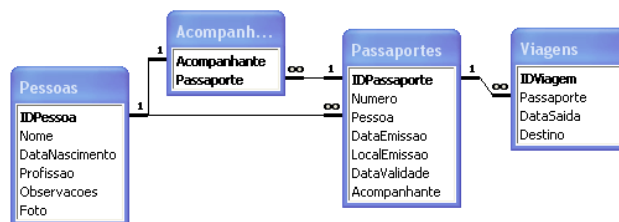


Figura 14.6: Diagrama Entidade-Relacionamento da base de dados dos Passaportes.

## 14.2 Passos para a Integração de Fontes Heterogêneas

A partir da definição de quais fontes de informação compõe o SIME – base de dados dos passaportes e documentos anotados XML para os almanaques e os diários de viagem – utiliza-se o Metamorphosis para se obter uma visão homogênea destas fontes heterogêneas.

O processo envolvendo a geração dessa visão é composto por uma série de etapas. As primeiras etapas requerem um especialista no universo de discurso em questão, pois ele será responsável por definir qual a visão que será oferecida aos utilizadores do sistema, ou seja, aos visitantes do Museu Virtual da Emigração. Esse especialista deve possuir sólidas noções sobre a norma *Topic Maps*, permitindo assim tornar a especificação o mais abrangente possível.

De forma resumida, as 3 etapas necessárias para atingir a visualização homogênea das fontes são:

1. Construção do topic map pelo Oveia (Capítulo 8) a partir das fontes heterogêneas. Para sua geração, seguem-se os 5 passos citados abaixo:
  - (a) Especificação das fontes de informação em XSDS;
  - (b) Definição da ontologia (dos conceitos e relacionamentos entre conceitos) do universo de discurso em causa;
  - (c) Especificação da ontologia em XS4TM;
  - (d) Relação dos conceitos contidos na ontologia com as fontes de informação (de onde se originam as informações referentes a cada tópico);
  - (e) Processamento das especificações XSDS e XS4TM, gerando o topic map a partir das fontes de informação;
2. Validação do topic map obtido através de uma especificação XTche (Capítulo 9);
3. Geração, recorrendo ao Ulisses (Capítulo 10), de um sítio Web que permite a navegação conceptual com uma visão homogênea dos recursos.

Nas próximas secções, descreve-se ao detalhe como foi realizado o processo para a obtenção da navegação conceptual referente ao Museu da Emigração.

## 14.3 Construção do topic map pelo Oveia

Esta secção tem por objectivo demonstrar, passo a passo, o processo de geração do topic map a partir dos dados extraídos das fontes heterogêneas de informação. O topic map gerado segue a ontologia descrita na especificação XS4TM que descreve o domínio do ME. Cada subsecção, a partir de agora, descreve um dos 5 passos citados no item 1 da secção anterior.

### 14.3.1 Especificação das fontes em XSDS

Esta etapa é responsável por definir a especificação XSDS que determina quais fontes de dados serão integradas e quais os dados que estão em cada fonte. Conforme apresentado na Secção 8.2, a especificação XSDS divide-se em duas etapas:

- Definição dos recursos físicos a serem interrogados;
- Selecção das partes de informação contidas nestes recursos necessárias para a construção da rede semântica.

Conforme anteriormente citado, o SIME envolve três fontes de dados: dois documentos XML (um referente aos Almanques e outro referente aos Diários de Viagem) e uma base de dados Microsoft Access® (referente aos Passaportes). Assim, a especificação em XSDS de cada fonte define-se em um elemento `<datasource>`, como abaixo representado:

```

1 <resources>
2   <datasources>
3     <!-- Declaração da Fonte de Dados dos Almanques -->
4     <datasource extratorDriver="br.uneb.dcet.tmbuilder.drivers.XMLFile" name="XML_Almanaque">
5       <parameter name="pathDocument">C:\\MUSEU\\Almanques\\Alm.xml</parameter>
6     </datasource>
7     <!-- Declaração da Fonte de Dados dos Diários de Viagem -->
8     <datasource extratorDriver="br.uneb.dcet.tmbuilder.drivers.XMLFile" name="XML_Viagem">
9       <parameter name="pathDocument">C:\\MUSEU\\DiariosViagem\\DViag.xml</parameter>
10    </datasource>
11    <datasource extratorDriver="br.uneb.dcet.tmbuilder.drivers.DataBase" name="BD_Passaporte">
12      <parameter name="connectionURL">jdbc:access://localhost/museu/passaporte</parameter>
13      <parameter name="password"/>
14      <parameter name="user"/>
15      <parameter name="jdbcDriver">sun.jdbc.odbc.JdbcOdbcDriver</parameter>
16    </datasource>
17  </datasources>
18  <datasets>
19    ...
20  </datasets>
21 </resources>

```

Para o acesso aos documentos XML, basta apenas informar o seu caminho na estrutura de directorias do sistema operativo. Aliado a isto, define-se o nome de cada *dataset* que será criado (no atributo *name*) e informa-se o *driver* responsável pela extracção (no atributo *extratorDriver*).

Para o acesso na base de dados relacional Microsoft Access® define-se um nome para o *dataset* a ser criado e do driver de extracção. Além disso, é necessário a criação de quatro parâmetros: a URL de conexão, o utilizador, sua palavra-passe e o *driver* JDBC.

Uma vez definidas as fontes de dados, parte-se para a definição de quais elementos (em documentos XML) ou campos (em base de dados) devem ser extraídos. Para isso, o elemento `<dataset>` é utilizado, o qual referencia qual fonte de dados (*datasource*) será a matéria-prima para a extracção.

```

1 <resources>
2   <datasources>
3     ...
4   </datasources>
5   <datasets>
6     <dataset name="Alm-Emigrante" database="XML_Almanaque">
7       <columns>
8         <column identify="true">@ID</column>
9         <column>nome</column>
10        <column>idade</column>
11        <column>profissao</column>
12      </columns>
13      <statement>//Notas_Biograficas/MEMORIA/PESSOA</statement>
14    </dataset>
15    <dataset>
16      ...
17    </dataset>
18    ...
19  </datasets>
20</resources>

```

O *dataset* acima apresentado efectua uma extracção no documento XML dos Almanagues. O conjunto de dados extraídos será referenciado pelo nome de *Alm-Emigrante* e conterá 4 elementos, referidos nos elementos `<column>`. Cada um desses elementos será encontrado fisicamente através do caminho XPath declarado no elemento `<statement>`, agregado ao conteúdo do elemento `<column>`.

Para exemplificar, o *dataset* chamado *Alm-Emigrante* formará uma tabela, onde cada linha conterá as seguintes informações extraídas da fonte *XML\_Almanaque*:

- `//Notas_Biograficas/MEMORIA/PESSOA/@ID` – será o identificador deste *dataset*, definido pelo atributo *identify*.
- `//Notas_Biograficas/MEMORIA/PESSOA/nome`
- `//Notas_Biograficas/MEMORIA/PESSOA/idade`
- `//Notas_Biograficas/MEMORIA/PESSOA/profissao`

No fim deste passo, todos os elementos encontrados no documento XML serão mapeados para este *dataset*, onde cada tuplo contém os dados referentes a um emigrante.

Porém, o conceito *Emigrante* encontra-se nas três fontes de informação do SIME. Por este motivo, deve-se fazer uma especificação para cada fonte. Assim sendo, para a fonte dos diários de viagem, o *dataset* dos emigrantes foi assim especificado:

```

1 <resources>
2   <datasources>
3     ...
4   </datasources>
5   <datasets>
6     <dataset name="DViagem-Emigrante" database="XML_DViagem">
7       <columns>
8         <column identify="true">@ID</column>
9         <column>nome</column>
10        <column>idade</column>
11        <column>profissao</column>
12      </columns>
13      <statement>//Notas_Bio/PESSOA</statement>
14    </dataset>
15    <dataset>
16      ...
17    </dataset>
18    ...
19  </datasets>
20</resources>

```

Pelo facto de que a estrutura das informações sobre um emigrante é similar nas duas fontes (almanaques e diários de viagem), a especificação XSDS para ambos é praticamente idêntica. Ressalta-se que a única diferença na declaração dos *datasets* é o conteúdo do elemento `<statement>`, pois o caminho para o elemento `<PESSOA>` não é o mesmo nas duas fontes.

O *dataset* a seguir define a extracção dos dados referentes às histórias de almanaques. Segue o mesmo formato do *dataset Alm-Emigrante*, definindo o caminho até o elemento principal (declarado em `<statement>`), o identificador (declarado em `<column identify="true">`) e os elementos que devem ser extraídos para o *dataset* em questão.

```

1   <datasets>
2     ...
3     <dataset name="Alm-HistoriaAlmanaque" database="XML_Almanaque">
4       <columns>
5         <column identify="true">@ID</column>
6         <column>localNasc</column>
7         <column>dataNasc</column>
8         <column>docAnexo</column>
9         <column>paisEmigrou</column>
10        <column>dataObito</column>
11        <column>dataCasamento</column>
12        <column>dataBaptizado</column>
13        <column>dataHomenagem</column>
14        <column name="pessoa">../PESSOA/@ID</column>
15      </columns>
16      <statement>//Notas_Biograficas/MEMORIA/HISTORIA_A</statement>
17    </dataset>
18    ...
19  </datasets>

```

Para o caso da definição do *dataset* extraído da base de dados dos passaportes (contendo o seu identificador, nome, nome do pai e nome de mãe), o seguinte código elemento `<dataset>` deve ser definido:

```

1   <dataset name="Passaporte-Emigrante" database="BD_Passaporte">
2     SELECT ID, Nome, NomePai, NomeMae FROM Pessoas

```

3 | </dataset>

Percebe-se, deste modo, que a linguagem de interrogação às fontes de dados varia de acordo com a própria fonte: quando a extracção é realizada sobre documentos XML, utiliza-se a linguagem XPath; quando a extracção é realizada sobre base de dados, utiliza-se a linguagem SQL.

### 14.3.2 Definição da ontologia do universo de discurso

Esta subsecção apresenta a definição da ontologia para o Museu da Emigração, contendo a representação de todo o conhecimento deste domínio.

A definição da ontologia consistiu, inicialmente, em determinar quais conceitos eram considerados importantes nesse domínio. Após este passo, determinou-se os relacionamentos entre tais conceitos.

Em termos práticos, num primeiro passo foi definido o conjunto de conceitos encontrados no domínio dos Almanques, Diários de Viagem e Passaportes. Esses conceitos estão listados alfabeticamente na Tabela 14.1:

Tabela 14.1: Conceitos encontrados no domínio do Museu da Emigração.

Acompanhante	Bens	Data	Data de Baptizado
Data de Casamento	Data de Emissão	Data de Homenagem	Data de Nascimento
Data de Óbito	Data de Saída	Data de Validade	Destino
Documento Anexo	Emigrante	História de Almanaque	História de Viagem
Idade	Local de Emissão	Local de Nascimento	Nota
País que Emigrou	Passaporte	Perfil	Profissão
Relações Sociais	Tipo de Perfil	Viagem	

O próximo passo consiste da definição dos relacionamentos entre os conceitos. Então, o especialista identifica quais conceitos estão relacionados e cria uma associação entre ambos.

Para exemplificar, tomamos como exemplo o caso entre os conceitos *Emigrante* e *Histórias de Almanaque*. Ao definir uma associação entre estes dois conceitos, fez-se necessário determinar como um conceito relaciona-se com o outro. Assim sendo, determina-se que um emigrante é referido numa história de almanaque.

Adaptando-se para a terminologia da norma **Topic Maps**, os nomes *é referido na história* é visto como papel de actuação na associação entre *Emigrante* e *Histórias de Almanaque*, pois expressa o ponto de vista de um conceito em relação ao outro.

No final da identificação de todas as associações, foi obtido um grafo representando a ontologia, conforme apresentado na Figura 14.7. Nesta figura, os nodos representam os conceitos, enquanto que as associações são representadas pelos arcos.

Essa ontologia representa o universo de discurso do Museu da Emigração, contendo os conceitos das diversas fontes de informação apresentados na Tabela 14.1.

Como o conceito *Emigrante* foi encontrado em todas as fontes, acabou por tornar-se um elo de união pois os conceitos *Emigrante* foram fundidos num único. Assim sendo, o conceito

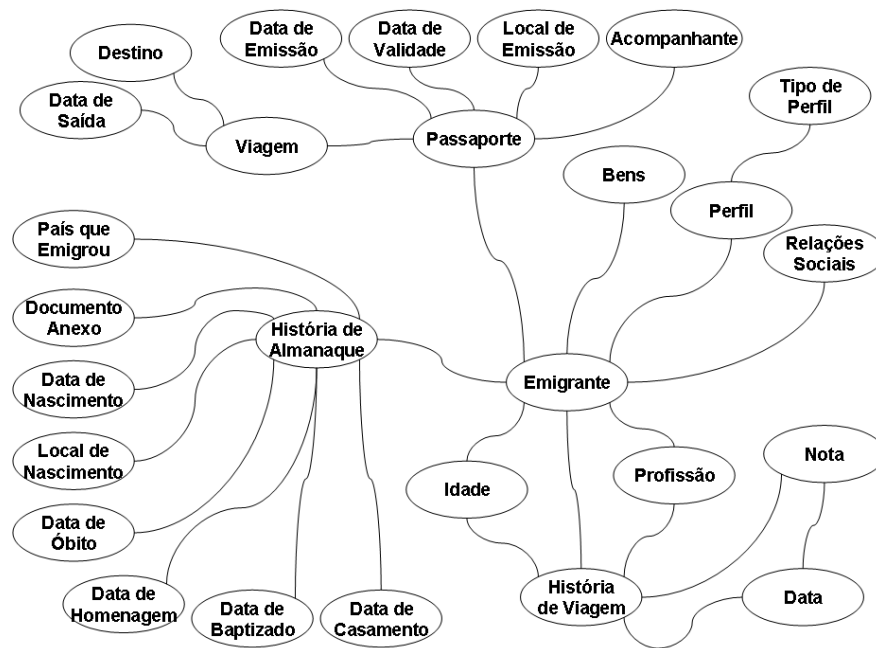


Figura 14.7: Ontologia para o Museu da Emigração

resultante mantém todos os relacionamentos que os demais participavam.

### 14.3.3 Especificação da Ontologia em XS4TM

A partir do momento que todos os conceitos e seus relacionamentos foram definidos, foi então especificado em XS4TM, a ontologia apresentada na subsecção anterior.

A primeira etapa da especificação em XS4TM consistiu na codificação dos tipos de tópicos (ou seja, os conceitos). Tomando como exemplo os tipos de tópicos *Emigrante* e *História de Almanaque*, especificou-se o seguinte código XS4TM:

```

1 <xs4tm xmlns="http://www.topicmaps.org/xtm/1.0/"
2   xmlns:xlink="http://www.w3.org/1999/xlink">
3   <ontologies>
4     <topic id="MuseuEmigracao">
5       <baseName>
6         <baseNameString>Museu da Emigração</baseNameString>
7       </baseName>
8     </topic>
9     <topic id="Emigrante">
10      <instanceOf>
11        <topicRef xlink:href="#MuseuEmigracao"/>
12      </instanceOf>
13      <baseName>
14        <baseNameString>Emigrante</baseNameString>
15      </baseName>
16    </topic>
17    <topic id="Historia-Almanaque">

```

```

18         <instanceOf>
19             <topicRef xlink:href="#MuseuEmigracao"/>
20         </instanceOf>
21         <baseName>
22             <baseNameString>História de Almanaque</baseNameString>
23         </baseName>
24     </topic>
25     ...
26 </ontologies>
27 <instances/>
28 </xs4tm>

```

No código acima, definiu-se os tópicos *Emigrante* e *História de Almanaque* como sendo instâncias do tópico *Museu da Emigração* (tópico com o identificador *MuseuEmigracao*), definindo-se uma estrutura hierárquica.

A partir da definição dos tipos de tópicos, parte-se para a definição dos tipos das associações. Assim como os anteriores, os tipos das associações são definidos como sub-elementos do elemento `<ontologies>`. Abaixo, encontra-se a especificação do tipo de associação que envolve os tópicos *Emigrante* e *História de Almanaque*.

```

1 <ontologies>
2     ...
3     <topic id="Emigrante-historiaAlmanaque">
4         <baseName>
5             <baseNameString>História de Almanaque de um Emigrante</baseNameString>
6         </baseName>
7         <baseName>
8             <scope>
9                 <topicRef xlink:href="#emigrante-referido-historia"/>
10            </scope>
11            <baseNameString>é referido na história</baseNameString>
12        </baseName>
13        <baseName>
14            <scope>
15                <topicRef xlink:href="#historia-refere-emigrante"/>
16            </scope>
17            <baseNameString>refere o Sr(a).</baseNameString>
18        </baseName>
19    </topic>
20    ...
21 </ontologies>

```

A leitura correcta desta associação seria: toda associação do tipo *História de Almanaque de um Emigrante* vai conter um emigrante que é referido em uma história, assim como conterá uma história que refere um emigrante.

No tópico acima destacam-se os papéis de actuação em associações (*association roles*) identificados por *emigrante-referido-historia* e *historia-refere-emigrante*. Estes papéis de actuação também são tópicos e devem ser encontrados declarados no elemento `<ontologies>`, conforme apresentado abaixo:

```

1 <ontologies>
2     ...
3     <topic id="emigrante-referido-historia"/>
4     <topic id="historia-refere-emigrante"/>

```



```

5 | ...
6 | </ontologies>

```

Ao final deste passo, a ontologia do Museu da Emigração estará definida no elemento `<ontologies>`. A mesma consiste basicamente da definição dos tipos de tópicos, dos tipos de associações e dos papéis de actuação em associações, conforme apresentado nessa subsecção.

Essa ontologia pode ser completada com os tipos e contextos dos nomes (*basenames*) e das ocorrências, assim como os contextos de associações que poderão ser encontrados na definição das instâncias:

```

1 | <ontologies>
2 |   <!-- Contextos -->
3 |   <topic id="Idade"/>
4 |   <!-- Tipos de basenames -->
5 |   <topic id="Profissao"/>
6 |   <topic id="NomePai"/>
7 |   <topic id="NomeMae"/>
8 |   ...
9 | </ontologies>

```

Estes tópicos podem conter características adicionais, como nomes, identificador de temas e ocorrências, caso seja necessário. Neste caso, em particular, decidiu-se pela suas declarações simplificadas (contendo apenas um identificador). Contudo, quanto mais informações agregam-se a tais tópicos, melhor a semântica será expressada.

#### 14.3.4 Instanciação da Ontologia em XS4TM

Uma vez definida a ontologia (Subsecção 14.3.3), o próximo passo consiste de relacioná-la com os *datasets* especificados em XSDS (Subsecção 14.3.1). Esta relação entre conceitos e fonte de informação é realizada no elemento `<instances>` da especificação XS4TM.

Em XS4TM, cada elemento `//instances/topic` e `//instances/association` possui um atributo chamado `dataset`, o qual determina de qual *dataset* os dados serão obtidos para a construção do tópico ou associação em questão.

Como exemplo, apresenta-se a seguir a especificação da extracção dos tópicos que possuem o tipo *Emigrante*. Neste exemplo percebe-se que o tópico instância de *Emigrante* está definido três vezes. Isso se justifica porque este conceito encontra-se em três fontes distintas: nos almanaques (*dataset Alm-Emigrante*, linha 4 a 23), nos diários de viagem (*dataset DViagem-Emigrante*, linha 24 a 43) e nos passaportes (*dataset Passaporte-Emigrante*, linha 44 a 63).

```

1 | <xs4tm xmlns="http://www.topicmaps.org/xtm/1.0/" xmlns:xlink="http://www.w3.org/1999/xlink">
2 |   <ontologies/>
3 |   <instances>
4 |     <topic dataset="Alm-Emigrante" id="@Alm-Emigrante.ID">
5 |       <instanceOf>
6 |         <topicRef xlink:href="#Emigrante"/>

```

```

7      </instanceOf>
8      <baseName>
9          <baseNameString>@Alm-Emigrante.nome</baseNameString>
10     </baseName>
11     <occurrence>
12         <scope>
13             <topicRef xlink:href="#Idade"/>
14         </scope>
15         <resourceData>@Alm-Emigrante.idade</resourceData>
16     </occurrence>
17     <occurrence>
18         <instanceOf>
19             <topicRef xlink:href="#Profissao"/>
20         </instanceOf>
21         <resourceRef xlink:href="@Alm-Emigrante.profissao"/>
22     </occurrence>
23 </topic>
24 <topic dataset="DViagem-Emigrante" id="@DViagem-Emigrante.@ID">
25     <instanceOf>
26         <topicRef xlink:href="#Emigrante"/>
27     </instanceOf>
28     <baseName>
29         <baseNameString>@DViagem-Emigrante.nome</baseNameString>
30     </baseName>
31     <occurrence>
32         <scope>
33             <topicRef xlink:href="#Idade"/>
34         </scope>
35         <resourceData>@DViagem-Emigrante.idade</resourceData>
36     </occurrence>
37     <occurrence>
38         <instanceOf>
39             <topicRef xlink:href="#Profissao"/>
40         </instanceOf>
41         <resourceRef xlink:href="@DViagem-Emigrante.profissao"/>
42     </occurrence>
43 </topic>
44 <topic dataset="Passaporte-Emigrante" id="@Passaporte-Emigrante.@ID">
45     <instanceOf>
46         <topicRef xlink:href="#Emigrante"/>
47     </instanceOf>
48     <baseName>
49         <baseNameString>@Passaporte-Emigrante.Nome</baseNameString>
50     </baseName>
51     <occurrence>
52         <instanceOf>
53             <topicRef xlink:href="#NomePai"/>
54         </instanceOf>
55         <resourceRef xlink:href="@Passaporte-Emigrante.NomePai"/>
56     </occurrence>
57     <occurrence>
58         <instanceOf>
59             <topicRef xlink:href="#NomeMae"/>
60         </instanceOf>
61         <resourceRef xlink:href="@Passaporte-Emigrante.NomeMae"/>
62     </occurrence>
63 </topic>
64 </instances>
65 </xs4tm>

```

No processo de extracção realizado pelo Oveia, será criado um tópico para cada emigrante encontrado nos *datasets* acima. Num segundo passo, o Oveia faz uma fusão entre os tópicos que possuem o mesmo identificador. Deste modo, se um mesmo emigrante é encontrado

nas três fontes de dados do SIME, um único tópico conterá toda a informação acerca do emigrante.

Esta especificação irá determinar a extracção de todos os tópicos do tipo *Emigrante*, os quais conterão um nome e até quatro ocorrências: idade, profissão (provenientes dos *datasets Alm-Emigrante* e *dataset DViagem-Emigrante*), nome do pai e nome da mãe (originadas do *dataset Passaporte-Emigrante*). Esse processo de associação dos tópicos com os *datasets* (como apresentado para o conceito *Emigrante*) deve ser realizado para todos os tipos de tópicos definidos na ontologia desse domínio.

Até o momento, o único tipo de associação entre tópicos existente são as relações classe-subclasse, definidas na hierarquia de tópicos. Portanto, no próximo passo será definido a extracção das associações para completar a especificação XS4TM.

Para exemplificar, usa-se a declaração de associações com o relacionamento entre *Emigrantes* e *Histórias de Almanaque*. O tipo dessa associação foi anteriormente declarado (`<topicid='Emigrante-historiaAlmanaque'>` na Subsecção 14.3.3), onde os nomes dos papéis de actuação foram definidos. A associação abaixo então relaciona os tais papéis com os tópicos que desempenharão cada função:

```

1      <association dataset="Alm-Emigrante">
2          <instanceOf>
3              <topicRef xlink:href="#Emigrante-historiaAlmanaque"/>
4          </instanceOf>
5          <member>
6              <roleSpec>
7                  <topicRef xlink:href="#emigrante-referido-historia"/>
8              </roleSpec>
9              <topicRef xlink:href="@Alm-Emigrante.@ID"/>
10         </member>
11         <member>
12             <roleSpec>
13                 <topicRef xlink:href="#historia-refere-emigrante"/>
14             </roleSpec>
15             <topicRef xlink:href="@Alm-HistoriaAlmanaque.pessoa"/>
16         </member>
17     </association>

```

Baseada nesta especificação de associações, será criado uma nova associação toda vez que os campos contidos nos *datasets* `@Alm-Emigrante.@ID` e `@Alm-HistoriaAlmanaque.pessoa` forem iguais. Na prática, quando isto ocorrer, indica que um emigrante está sendo citado numa história.

A cardinalidade, neste caso, será definida pelas fontes de dados; ou seja, se um mesmo emigrante está referido em várias histórias de almanaques, será criada uma associação para cada referência encontrada nas fontes. Assim, um mesmo emigrante pode ser citado em várias histórias de almanaque.

### 14.3.5 Extracção do topic map com o Oveia

O processo de extracção por intermédio do Oveia foi realizado em 4 etapas:

1. abertura do projecto de extracção, escolhendo as especificações nas linguagens XSDS e XS4TM a processar;
2. configuração do formato de saída do topic map gerado (ou em uma base de dados ou em um documento XML);
3. verificação das informações estatísticas dos elementos a serem extraídos;
4. processamento propriamente dito da extracção e visualização dos resultados da extracção.

Na Etapa 1 são definidas as informações referentes ao projecto de extracção, conforme apresenta a Figura 14.8. Nesta etapa, pode-se escolher um projecto existente para reprocessar a extracção (caso o mesmo já tenha sido realizado anteriormente) ou criar um novo projecto.

Ao escolher um novo projecto, é necessário informar o nome, autor, descrição e seleccionar os dois ficheiros de especificação: XSDS e XS4TM.

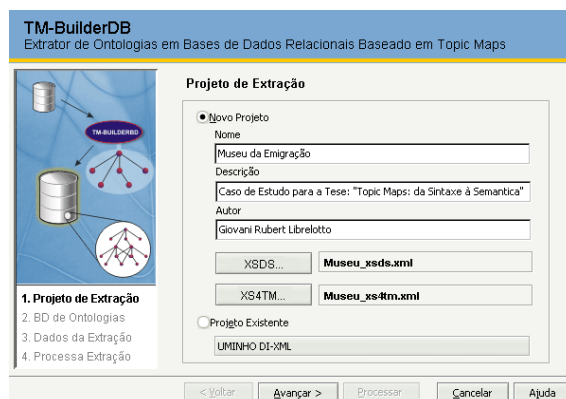


Figura 14.8: Etapa 1 – O Projecto de Extracção.

Após definir o projecto, definem-se as configurações de saída do Oveia, na Etapa 2. A Figura 14.9 apresenta 2 imagens, referentes à escolha (alternativas não-exclusivas) do armazenamento do topic map na BD Ontologia (imagem à esquerda) e da exportação do topic map para um ficheiro XTM (imagem à direita).

Para a especificação do armazenamento do topic map na BD Ontologia, no item *Plataforma* selecciona-se qual é o tipo da base de dados a ser criada; por exemplo, SQL Server, MySQL, Oracle, etc. A seguir, insere-se os **drivers** de conexão com a base de dados, o protocolo utilizado, o endereço do servidos da base de dados, o utilizador e a palavra-passe necessárias para o estabelecimento da conexão.

Para a especificação da criação do topic map num ficheiro XTM, basta seleccionar a opção *XML file* no item *Plataforma*. O topic map resultante será criado na directoria por omissão (*default*) do Oveia.

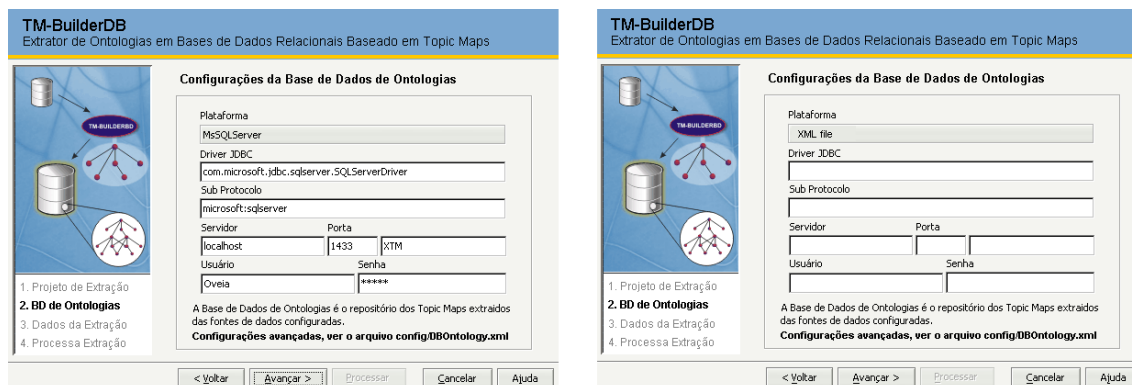


Figura 14.9: Etapa 2 – Selecção do modo de armazenamento do topic map: na BD Ontologia (esquerda) ou num ficheiro XTM (direita)

A Etapa 3 (Figura 14.10) fornece uma breve estatística da previsão do total de elementos que deverão ser extraídos a partir do projecto seleccionado. Esses dados correspondem às informações obtidas da especificação XS4TM.

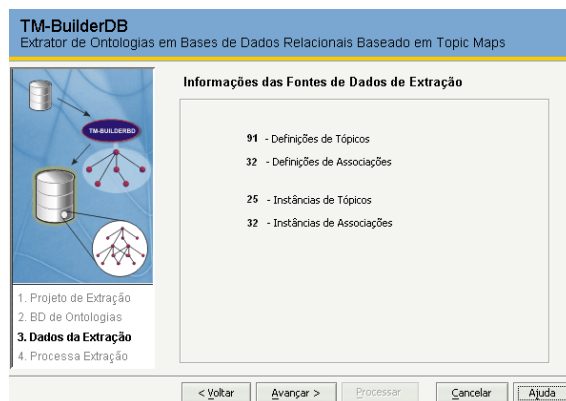


Figura 14.10: Etapa 3 – Dados da extracção

A Etapa 4 consiste no processo de extracção propriamente dito. Nesta etapa realiza-se a execução da extracção. Após o término do processamento, uma estatística completa referente ao processo é fornecida, conforme apresenta a Figura 14.11, onde é informado o tempo de execução de todo processo, os elementos extraídos e o número de elementos que não foram extraídos, devido a alguma falha.

Ao final de todo o processamento, foi obtido um ficheiro de acordo com a sintaxe XTM com 35588 linhas, contendo a especificação de 1043 tópicos, 25 tipos de tópicos, 1541 associações e 32 tipos de associações, com um tamanho final de 1,14 MB (1.197.322 bytes). Por sua vez, a BD Ontologia armazenou o topic map gerado em suas tabelas, o qual contem o mesmo número de tópicos e associações que a versão XTM.

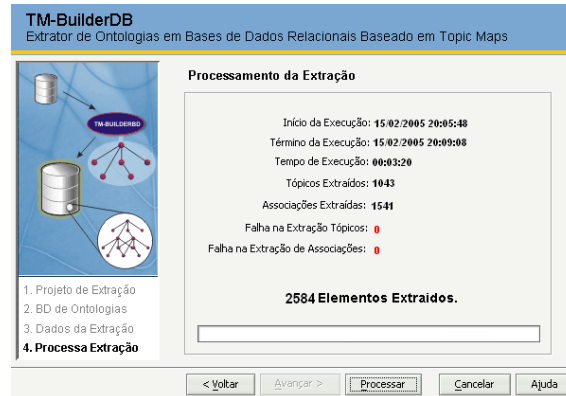


Figura 14.11: Etapa 4 – Resultados da Extração

## 14.4 Validação do topic map com XTche

Tomando como partida o topic map obtido após o processamento do Oveia, o passo seguinte se compõe da sua validação de acordo com um conjunto de regras definidas pelo especialista do domínio. A Figura 14.12 apresenta a especificação XTche referente aos tipos de tópico *Emigrante*, *História de Vida* e *História de Almanaque*.

De acordo com a definição do caso de estudo, todo tópico do tipo *Emigrante* deve possuir obrigatoriamente um nome e, opcionalmente, ter ocorrências dos tipos *profissão* e *observação* (*obs*). Sendo assim, o elemento *Emigrante* possui os sub-elementos `xtche:baseName` (indicando que haverá um nome) e os elementos *profissão* e *obs* com o atributo `xtche:occurrenceType`, indicando que ambos serão tipos de ocorrências que podem ser encontrados nos tópicos deste tipo, pois ambos são opcionais (cardinalidade entre 0 e 1).

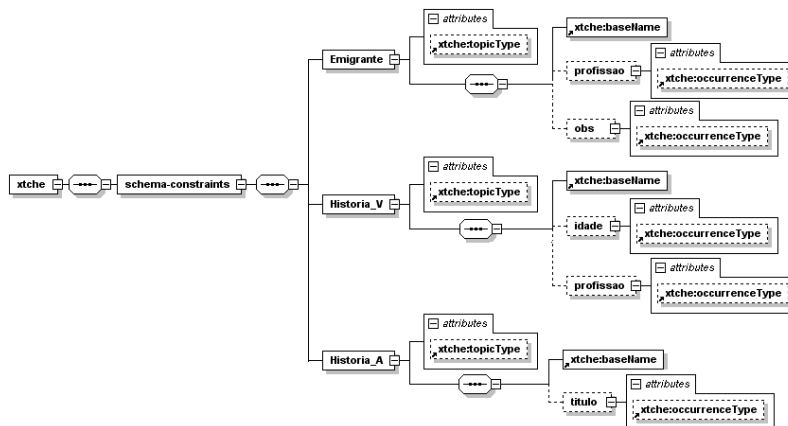


Figura 14.12: Restrições de esquema XTche para os tipos de tópico *Emigrante*, *História de Vida* e *História de Almanaque*

As restrições de estrutura dos tipos de tópicos *História de Vida* (*Historia\_V*) e *História de Almanaque* (*Historia\_A*) seguem o formato da regra definida para o tipo de tópico

*Emigrante*, conforme pode ser visto na Figura 14.12.

Os tópicos do tipo *História de Vida* devem possuir um nome e, opcionalmente, ocorrências do tipo *idade* e *profissão*. Essas ocorrências referem-se à idade e profissão do emigrante citado na história de vida em questão, permitindo que sejam contadas histórias de um mesmo emigrante em tempos distintos de sua vida.

Por sua vez, os tópicos do tipo *História de Almanaque* possuem uma estrutura composta por um nome e uma ocorrência opcional do tipo *título*. As demais informações sobre cada história de almanaque são definidas por associações envolvendo outros tópicos, como por exemplo: tópicos do tipo *data de nascimento*, *data de óbito*, *país que emigrou*, *local de nascimento*, etc<sup>2</sup>.

Além dos 3 tipos de tópicos mostrados na Figura 14.12, a especificação XTche define a estrutura de outros 22 tipos. A maioria destes possui uma estruturação simples, contendo apenas um nome, como é o caso do tipo de tópico *data de validade* e *destino*.

Em termos das restrições contextuais, define-se quais os tópicos que devem ser usados apenas como papéis de actuação, como por exemplo, os tópicos *faleceu no dia* (encontrado na associação entre emigrante e data de óbito), *teve como destino* (encontrado na associação entre passaporte e destino, indicando a viagem de um emigrante) e *é referido na história* (encontrado na associação entre emigrante e história de almanaque). No total, são 65 tópicos que são usados exclusivamente como papel de actuação em associações, que são especificados em XTche como um sub-elemento de `contextual-constraints` contendo o atributo `xtche:associationRole-Exclusive`.

Para definir as regras que devem ser obedecidas pelas associações, usar-se-á o exemplo das associações que envolvem justamente os tipos de tópicos definidos na Figura 14.12. A Figura 14.13 descreve o tipo de associação *Emigrante-Historia\_V* especifica que toda associação deste tipo obrigatoriamente possui um tópico do tipo *Emigrante* desempenhando um papel de actuação *possui-historia* e um tópico do tipo *Historia\_V* desempenhando um papel de actuação *relata-historia*.

O processo de especificação é idêntico para a associação entre *Emigrante* e *Historia de Almanaque* (*Historia\_A*), com excepção dos papéis de actuação. Neste caso, define-se que o papel *é referido numa história* é desempenhado por um tópico do tipo emigrante, enquanto que um tópico do tipo *história de almanaque* desempenha o papel *refere o emigrante*.

Para o caso de estudo do ME, foram definidos 32 tipos de associações envolvendo os tópicos deste domínio. Para todos os casos, definiu-se que a cardinalidade de cada associação é 1x1; ou seja, supondo que um emigrante seja referenciado em várias histórias de almanaque, será criado uma associação para definir cada relação.

O facto do topic map ter sido criado por uma ferramenta de construção automática reduz o número de restrições que possam detectar erros, pois certos casos já são verificados pelo próprio processo de criação. Isso quer dizer que todas as associações geradas pelo Oveia vão seguir a estrutura definida em XS4TM. Por exemplo, as associações que envolvem os

<sup>2</sup>As associações foram definidas baseadas na ontologia do ME, apresentada graficamente na Figura 14.7.

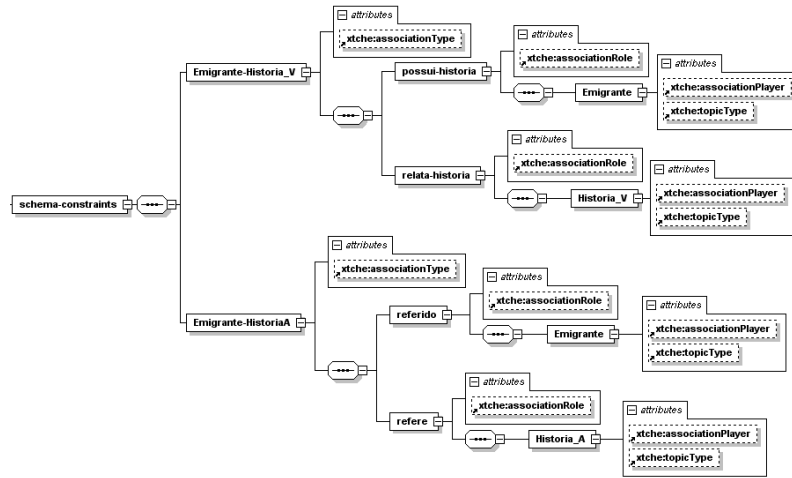


Figura 14.13: Restrições de esquema XTche para os tipos de associação que envolvem os tópicos Emigrante, História de Vida e História de Almanaque

tipos de tópicos *emigrantes* e *bens* sempre possuirão os papéis ideais sendo desempenhados por tópicos de um mesmo tipo, conforme apresenta a definição de extracção de associação abaixo:

```

1      <association dataset="Emigrantes">
2          <instanceOf>
3              <topicRef xlink:href="#id_Emigrante-Bem"/>
4          </instanceOf>
5          <member>
6              <roleSpec>
7                  <topicRef xlink:href="#id_possui"/>
8              </roleSpec>
9              <topicRef xlink:href="@Emigrantes.bens"/>
10         </member>
11         <member>
12             <roleSpec>
13                 <topicRef xlink:href="#id_pertence"/>
14             </roleSpec>
15             <topicRef xlink:href="@Bens.bens"/>
16         </member>
17     </association>

```

De acordo com a definição acima, toda associação do que envolva estes tópicos será composta por dois papéis de actuação, onde o papel possui é desempenhado por um tópico do tipo *emigrante*, enquanto que um tópico do tipo *bens* desempenha o papel *pertence*.

Desta forma, o Oveia cria este tipo de associação (assim como qualquer outro tipo definido em XS4TM) seguindo uma estrutura rígida – os mesmos tipos de tópicos desempenhando os mesmos papéis. Assim sendo, a validação XTche irá apenas confirmar que o Oveia realizou seu processo de uma maneira correcta, pois nenhum erro deste tipo é detectado pelo TM-Validator.



## 14.5 Navegação no topic map com Ulisses

A partir do topic map gerado pelo Oveia (Secção 14.3) e validado pelo XTche (Secção 14.4), o Ulisses se encarrega de produzir o website referente ao domínio do ME. A Figura 14.14 apresenta a página inicial da navegação, a qual segue o modelo apresentado na Subsecção 10.2.1 (versão I, estática, da ferramenta em causa, que gera um conjunto fixo de páginas HTML a partir de ficheiros XTM).



Figura 14.14: Página principal do website do Museu da Emigração

A integração das fontes é percebida na visualização das informações referentes a um emigrante. A Figura 14.15 apresenta a página HTML correspondente ao emigrante *Sérgio Carvalho*. Além da ocorrência textual do tipo *Profissão*, o emigrante em questão está envolvido em 27 associações, as quais são de 12 tipos distintos. Essas associações foram definidas a partir dos dados extraídos das 3 fontes. Por exemplo:

- a associação “*Sérgio Carvalho é citado em uma nota com a data 1985-08-13*” é oriunda de um diário de viagem, onde uma nota escrita no dia *1985-08-13* citava o emigrante *Sérgio Carvalho*;
- a associação “*Sérgio Carvalho é considerado alto, forte e barrigudo*” tem origem num almanaque;
- a associação “*Sérgio Carvalho tinha o passaporte CM420589*” surgiu dos dados obtidos a partir da base de dados dos passaportes.

Analisando desta maneira, toda a informação referente a este emigrante está apresentada de uma forma homogênea, pois para o utilizador que está a navegar no website pouco importa de qual fonte vieram os dados responsáveis por definir uma associação de um conceito; o importante é obter toda a informação disponível nas fontes de dados sobre o conceito desejado.

A página referente a um emigrante é o melhor exemplo de uma integração de fontes heterogêneas, pois como pode-se observar na Figura 14.7 (referente à ontologia sobre o domínio

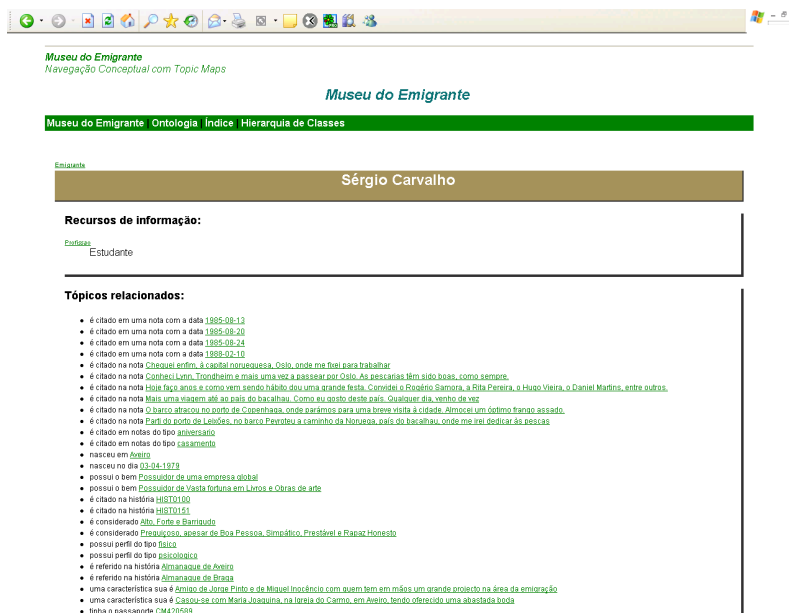


Figura 14.15: Página sobre um emigrante

do ME), o conceito *Emigrante* está relacionado com histórias de almanaques, passaportes e histórias de viagens, ou seja, todas as fontes de dados podem fornecer informação sobre o mesmo emigrante.

Cada associação encontrada na Figura 14.15 é composta por um papel de actuação seguida do conceito associado. Este conceito possui um *link* para a página que o descreve. Para exemplificar, a última associação diz que o emigrante *Sérgio Carvalho* tinha o passaporte *CM420589*. O nome *CM420589* na verdade é um *link* para a página referente ao próprio passaporte, a qual está apresentada na Figura 14.16.



Figura 14.16: Visão de um Passaporte.

O passaporte descrito na Figura 14.16 está envolvido em 8 associações de 6 tipos distintos. A partir destes relacionamentos pode-se obter informações como, por exemplo, a sua data de emissão (*16-12-2000*), o local de sua emissão (*Braga*) e o destino de uma viagem feita com ele (*Colômbia*).

Através dos *links* que descrevem cada associação, pode-se navegar conceptualmente através do grafo formado pelo topic map do ME. Portanto, cada *link* representa um arco que conecta dois nodos relacionados (as páginas que descrevem os conceitos).

## 14.6 Sumário do caso de estudo do Museu da Emigração

O caso de estudo em questão permitiu aplicar o *Metamorphosis* na combinação de fontes heterogêneas de dados que alimentam o Sistema de Informação do Museu da Emigração (SIME), de modo a facultar aos possíveis visitantes uma navegação sobre os dados relevantes extraídos das fontes de uma forma transparente e homogênea.

Em termos práticos, foram utilizados os três módulos do *Metamorphosis* para a obtenção da finalidade pretendida:

- o *Oveia* extraiu os dados contidos nas fontes heterogêneas de informação (uma base de dados de passaportes e dois documentos anotados XML de almanaques e diários de viagem), construindo um topic map baseado numa especificação *XS4TM*;
- o topic map gerado foi validado a partir de um conjunto de restrições especificadas em *XTche*;
- o *Ulisses* gerou um website a partir do topic map validado, no qual se pode navegar sobre os recursos do SIME.

A representação do conhecimento do domínio do Museu da Emigração está então descrita num topic map que contém a rede semântica formada pelos conceitos e os seus relacionamentos. Este topic map pode ser intercambiado posteriormente com outras aplicações e ferramentas, pois a sintaxe *XTM* é a mais difundida e aceite pela comunidade académica para a representação de *Topic Maps*.

O intercâmbio do topic map gerado com outras aplicações pode vir a permitir que o conhecimento acerca deste universo de discurso possa ser integrado com outros topic maps que representem domínios relacionados ao Museu da Emigração, aumentando assim a riqueza semântica da rede resultante.

Igualmente importante, é o facto de se poder gerar uma vista completamente diferente dos dados retirados das mesmas fontes, bastando para isso reformular a ontologia a representar, definida em *XS4TM* (Subsecção 14.3.2).



## Capítulo 15

# Conclusão

*Já não tenho mais assunto pra escrever  
Cumprimentos ao nosso pessoal  
Um abraço deste que tanto vos quer  
Sou capaz de ir aí pelo Natal*

João Monge

Com o objectivo principal de definir uma linguagem que permitisse a especificação de restrições sobre **Topic Maps**, a presente dissertação **Topic Maps: da Sintaxe à Semântica** apresentou a linguagem XTche.

XTche foi idealizada para suprir uma carência encontrada na norma ISO 13250 **Topic Maps** (Biezunsky, Bryan, and Newcomb, 1999). Conforme visto ao longo desta dissertação, **Topic Maps** é um formalismo para representar conhecimento descrito, ou contido, em um conjunto de recursos de informação, organizando-o em tópicos. Esses tópicos são relacionados através de associações. A informação sobre os tópicos pode ser inferida ao examinar as associações e ocorrências nos recursos de informação. Uma colecção desses tópicos e associações, isto é, uma rede semântica por elas formada é chamada topic map.

Permitindo que qualquer domínio de aplicação pudesse ser representado num topic map, passou a existir uma preocupação com a consistência dos topic maps criados. Assim, a ISO – através do grupo chamado ISO/IEC JTC 1/SC34, que é responsável pela normalização de **Topic Maps** – definiu um conjunto de requisitos que se deveriam poder validar em **Topic Maps**, dando origem àquilo que se chama TMCL – *Topic Map Constraint Language* – que na realidade é uma meta-linguagem.

O projecto da linguagem XTche baseou-se nos requisitos de TMCL, satisfazendo as exigências impostas pela ISO, classificando as restrições em **Topic Maps** em 3 tipos: restrições de esquema, restrições contextuais e restrições de existência.

Essa classificação trouxe um importante benefício: permite a especificação de esquemas para **Topic Maps** de uma mesma família. As restrições de esquema propiciam a definição das regras que uma classe de tópicos ou de associações devem seguir. Deste modo, com

um processador especializado, tornou-se possível gerar automaticamente um esqueleto de topic map que segue o esquema, ou seja, gera um documento com a estrutura de todos os tópicos de modo a poder ser posteriormente preenchido, manual ou automaticamente, para formar uma instância completa.

Visando facilitar o processo de especificação, XTche permite a validação semântica de topic maps a partir de uma linguagem apoiada na sintaxe XML Schema. A partir da visualização gráfica dos elementos e atributos, obtida com um editor de XML Schema, a estrutura das especificações XTche pode ser analisada, permitindo uma verificação visual. Além disso, utilizando a sintaxe de XML Schema, XTche adopta uma sintaxe que é uma recomendação W3C.

Outro objectivo desta dissertação foi a apresentação de um ambiente – *Metamorphosis* – capaz de automatizar o processo de construção de Topic Maps a partir de fontes heterogêneas de informação, possibilitando sua validação semântica e também a navegação sobre o conhecimento nele representado.

A principal ideia do *Metamorphosis* é integrar a especificação de redes de conceitos ou ontologias, com a sua extracção automática e a sua validação, assim como o seu armazenamento e a sua navegação, a partir de recursos heterogêneos de informação.

O funcionamento do *Metamorphosis* é composto por três módulos independentes, os quais podem ser utilizados em conjunto, sendo um deles obviamente o validador XTche.

A construção de Topic Maps é realizada pelo Oveia (apresentado no Capítulo 8), um processador que segue a ontologia especificada na linguagem XS4TM para extrair um topic map de fontes definidas na linguagem XSDS. Sucessor do TM-Builder (um extractor de Topic Maps que manipula exclusivamente fontes XML, descrito no Capítulo 7), o Oveia tem na linguagem XSDS seu factor diferencial, pois essa linguagem permite que várias fontes (como base de dados e documentos XML) possam servir como fontes de dados para a construção de topic maps. Além disso, a linguagem XS4TM cobre todos os elementos do modelo de dados da norma ISO 13250 Topic Maps (Garshol et al., 2003).

Como dito anteriormente, um topic map permite a representação de conhecimento sobre um domínio. Para isso, é necessário processadores que possibilitem exibir a informação que está nele representado. Para isso, desenvolveu-se o Ulisses: um construtor de navegadores conceptuais para Topic Maps (detalhado no Capítulo 10). A partir de um topic map, o Ulisses gera um conjunto de páginas HTML que apresentam a informação pertinente (sobre os tópicos e as associações) e que graças aos hiperlinks, permite navegar entre elas. Possuindo três variantes, o Ulisses permite a navegação em topic maps representados na sintaxe XTM ou em uma BD Ontologia, um formato relacional para Topic Maps.

Testado com sistemas de informação de pequeno, médio e grande porte, o *Metamorphosis* mostrou-se capaz de cumprir com seus objectivos, permitindo a extracção e construção de topic maps a partir de fontes heterogêneas de informação, validando-os semanticamente e gerando uma navegação conceptual a partir do universo de discurso pretendido, tal foi discutido nos 4 capítulos da terceira parte da dissertação.

## 15.1 Contributos

O principal contributo deste trabalho de doutoramento é, de facto, a linguagem XTche e seu ambiente de processamento. Além de XTche, é possível citar outros contributos deste doutoramento, os quais estão abaixo listados:

**Metamorphosis:** um ambiente de processamento de Topic Maps, propiciando extracção automática, validação e geração de navegadores conceptuais sobre um universo de discurso;

**TM-Builder:** um extractor de Topic Maps a partir de documentos XML a partir de uma especificação XSTM, criando uma ferramenta toda baseada na plataforma XML;

**Oveia:** um extractor de Topic Maps a partir de recursos de informação heterogéneos, baseado em especificações XSDS e XS4TM;

**BD Ontologia:** um modelo relacional para o armazenamento de Topic Maps.

Em termos de produção científica, de Maio de 2002 até Agosto de 2005 foram 25 artigos publicados em 9 países (Portugal, Brasil, Canadá, Estados Unidos, Espanha, Peru, Bolívia, Uruguai e Chile), sendo 2 deles seleccionados como *Best Papers* e publicados em edições especiais de um *journal*. Desses, 8 seguem na secção de referências bibliográficas desta dissertação, podendo as restantes ser encontradas no *Curriculum Vitae* do autor ou na página WWW do projecto em <http://www.di.uminho.pt/~gepl/metamorphosis/>. Além disso, foram realizados 2 tutoriais em conferências sobre *Representação do Conhecimento e Semantic Web*.

## 15.2 Trabalhos Futuros

A sequência que se idealiza para este trabalho de doutoramento é a submissão da linguagem XTche ao comité ISO/IEC JTC 1/SC34 e a consequente participação na elaboração da norma TMCL. O processo de ingresso no comité possui certas complicações burocráticas, pois para o membro ter direito a voto, deve estar representando um país. Portanto, um dos objectivos é tentar ser o representante brasileiro no comité (e paralelamente português, uma vez que Portugal também não tem representante) a fim de contribuir com a edição e definição da norma ISO 19756 Topic Maps Constraint Language (TMCL).

Em termos práticos, pretende-se aplicar a linguagem XTche a demais casos reais de aplicação, tal como o Museu da Emigração (apresentado no Capítulo 14), mostrando o ganho obtido com o uso dessa linguagem.

Infelizmente, não há uma forma explícita para impor restrições semânticas no momento da edição de um topic map real; o projectista deve evitar o uso de certos tópicos em sítios não adequados. Face a isso, pretende-se que a linguagem XTche possa vir a preencher essa lacuna, num futuro próximo, vindo a propiciar a criação de interfaces para a edição de Topic

**Maps** baseada no conjunto de regras numa especificação **XTche**. Estas interfaces serão construídas por um processador específico a partir das regras definidas. Assim, os erros contextuais serão verificados em tempo de edição; ou seja, caso o utilizador desobedeça uma das condições impostas pelas restrições no acto da edição do topic map, receberá mensagens de notificação quanto ao desrespeito por alguma dessas regras.

A fim de permitir que as especificações **XTche** possam ser interpretadas pelos processadores das demais linguagens para especificação de restrições em **Topic Maps**, está em projecto um tradutor que transforme especificações **XTche** em código **AsTMa!** ou **OSL**. O mesmo processador também teria a capacidade de transformar um documento com as regras de acordo com a sintaxe das demais linguagens para a sintaxe **XTche**, proporcionando a escolha da sintaxe e processador a utilizar para a validação de **Topic Maps**.

No âmbito do **Oveia**, um dos projectos a ser desenvolvido será um módulo que permita o armazenamento numa **BD Ontologia** de topic maps na sintaxe **AsTMa!** e **LTM**, assim como o processo inverso (descrição do conteúdo de uma **BD Ontologia** em **AsTMa!** ou **LTM**). Com este módulo, poderá se obter rapidamente um conjunto de **Topic Maps** armazenados no formato desejado: nas linguagens **XTM**, **AsTMa!** ou **LTM**, ou ainda numa **BD Ontologia**.

O **Oveia** não possui um ambiente amigável para o utilizador criar suas especificações. Actualmente, o processo de criação de um documento **XSDS** ou **XS4TM** é trabalhoso e maçador, pois exige-se que o projectista conheça o esquema destas linguagens. Por outro lado, essa tarefa pode ser auxiliada por ferramentas de edição de documentos **XML**, como exemplo o **XMLSpy**<sup>1</sup>, devido ao facto que elas permitem a validação estrutural da especificação em tempo de escrita.

Visando tornar o processo de especificação **XSDS** mais simplificado, planeia-se a construção de um módulo que tenha a capacidade de abstrair os relacionamentos encontrados nas bases de dados, a partir dos metadados das próprias bases. Essa abordagem usaria a estratégia de engenharia reversa, construindo automaticamente o documento de especificação **XSDS**. Partindo desta especificação, o projectista pode adoptá-la imediatamente ou efectuar as modificações de acordo com a sua necessidade.

No que diz respeito a trabalhos futuros relacionados ao **Ulisses**, está a ser estudado um componente que irá permitir ao utilizador especificar o aspecto visual do website gerado (por enquanto, apenas há uma configuração da interface). Desta forma, a visualização da ontologia representada no topic map visualizado pode ser apresentada da forma mais propícia, de acordo com a necessidade de cada topic map.

De uma forma similar, outro incremento a ser incorporado ao **Ulisses** será a possibilidade de filtrar o topic map no momento da navegação. Isto poderá ser utilizado nos casos em que o topic map é demasiado grande e o utilizador quer aceder à informação de um determinado contexto ou referente a um domínio em específico. Assim evita-se lidar com todo o topic map; o **Ulisses** apenas se preocuparia em carregar os tópicos e associações que realmente são importantes para a referida navegação.

---

<sup>1</sup>Ferramenta para construção de documentos **XML**, desenvolvido pela **ALTOVA**. Mais informações em: <http://www.altova.com>



Ainda no âmbito do *Ulisses*, outra família de navegadores está em desenvolvimento: navegadores gráficos para *Topic Maps* baseados em SVG (Ferraiolo and Jackson, 2003) e X3D (Web3D, 2004). Estes navegadores serão incorporados ao *Metamorphosis*, tornando-o uma ferramenta cada vez mais completa, no que diz respeito à norma *Topic Maps*.

Por fim, esse trabalho de doutoramento não abordou o tema fusão (*merge*) de *Topic Maps*. Um dos trabalhos que podem vir a ser realizados é o processo de fusão de topic maps armazenados na BD Ontologia. Outro trabalho que pode ser desenvolvido é a fusão de especificações XSDS e XS4TM, integrando representações de universos de discurso distintos.



# Bibliografia

- [Abiteboul, Buneman, and Suciu1999] Abiteboul, Serge, Peter Buneman, and Dan Suciu. 1999. *Data on the Web : From Relations to Semistructured Data and XML*. Morgan Kaufmann Series in Data Management Systems, 1st edition.
- [Ahmed et al.2001] Ahmed, Kal, Danny Ayers, Mark Birbeck, Jay Cousins, David Dodds, Joshua Lubell, Miloslav Nic, Daniel Rivers-Moore, Andrew Watt, Rob Worden, and Ann Wrightson. 2001. *Professional XML Meta Data*. Wrox Programmer to Programmer Series.
- [Albrecht1999] Albrecht, Jochen. 1999. Geospatial information standards a comparative study of approaches in the standardisation of geospatial information. *Computers and Geosciences*, 25(1):9–24.
- [Algermissen2002] Algermissen, Jan. 2002. Mapalizer. <http://www.jalgermissen.com/mapalizer/>.
- [Apache2003] Apache. 2003. Apache Jakarta Tomcat. The Apache Software Foundation. <http://jakarta.apache.org/tomcat/>.
- [Apache2005] Apache. 2005. Jelly : Executable XML. The Jakarta Project, Apache Software Foundation. <http://jakarta.apache.org/commons/jelly/>.
- [Ausubel1968] Ausubel, David P. 1968. *Educational Psychology, A Cognitive View*. New York: Holt, Rinehart and Winston, Inc.
- [Barta2002] Barta, Robert. 2002. XTM::LTM - Topic Map Parsing of LTM instances. <http://search.cpan.org/dist/XTM/lib/XTM/LTM.pm>.
- [Barta2003a] Barta, Robert. 2003a. AsTMa! Bond University, TR. <http://astma.it.bond.edu.au/constraining.xsp>.
- [Barta2003b] Barta, Robert. 2003b. AsTMa\* Language Family. Bond University, TR. <http://astma.it.bond.edu.au/astma-family.dbk>.
- [Barta2004] Barta, Robert. 2004. AsTMa= Language Definition. Bond University, TR. <http://astma.it.bond.edu.au/astma=-spec-xtm.dbk>.

- [Bechhofer et al.2004] Bechhofer, Sean, Jim Hendler, Ian Horrocks, Peter F. Patel-Schneider, and Lynn Andrea Stein. 2004. Web Ontology Language (OWL) – Reference. <http://www.w3.org/TR/owl-ref/>, February.
- [Bechhofer et al.2002] Bechhofer, Sean, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah McGuinness, Peter Patel-Schneider, and Lynn Andrea Stein. 2002. Web Ontology Language (OWL) Reference Version 1.0. World Wide Web Consortium, November. <http://www.w3.org/TR/owl-ref/>.
- [Beckett and McBride2004] Beckett, Dave and Brian McBride. 2004. RDF/XML Syntax Specification (Revised). World Wide Web Consortium, February. <http://www.w3.org/TR/rdf-syntax-grammar/>.
- [Berners-Lee2000] Berners-Lee, Tim. 2000. W3C – Semantic Web – XML 2000. <http://www.w3.org/2000/Talks/1206-xml2k-tbl/>.
- [Berners-Lee, Hendler, and Lassila2001] Berners-Lee, Tim, James Hendler, and Ora Lassila. 2001. The Semantic Web. In *Scientific American*. <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>, May.
- [Biezunski and Newcomb1999] Biezunski, Michel and Steven R. Newcomb. 1999. Topic Maps Frequently Asked Questions, September. <http://www.infoloom.com/faq.htm>.
- [Biezunsky, Bryan, and Newcomb1999] Biezunsky, Michel, Martin Bryan, and Steve Newcomb. 1999. ISO/IEC 13250 - Topic Maps. ISO/IEC JTC 1/SC34, December. <http://www.y12.doe.gov/sgml/sc34/document/0129.pdf>.
- [Biezunsky, Bryan, and Newcomb2003] Biezunsky, Michel, Martin Bryan, and Steve Newcomb. 2003. Summary of Voting on SC 34 N0358 Restatement of Topic Maps. ISO/IEC 13250:2002 ISO/IEC JTC1 SC34 N0388. <http://www.y12.doe.gov/sgml/sc34/document/0388.htm>.
- [Biron2001] Biron, Paul V. 2001. XML Schema part 2: Datatypes. <http://www.w3.org/TR/xmlschema-2>, May.
- [Booch, Rumbaugh, and Jacobson1999] Booch, Grady, James Rumbaugh, and Ivar Jacobson. 1999. *The Unified Modeling Language User Guide*. Reading, MA: Addison-Wesley.
- [Bray, Hollander, and Layman1999] Bray, Tim, Dave Hollander, and Andrew Layman. 1999. Namespaces in XML. World Wide Web Consortium. <http://www.w3.org/TR/REC-xml-names/>.
- [Brickley2000] Brickley, Dan. 2000. *RDF Specifications: Containing Resource Description Framework RDF Schema and Resource Description Framework RDF Model and Syntax Specification*. iUniverse.com, October.
- [Brickley and Guha2000a] Brickley, Dan and Ramanathan V. Guha. 2000a. Resource Description Framework (RDF) Schema Specification 1.0. World Wide Web Consortium, March. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>.

- [Brickley and Guha2000b] Brickley, Dan and Ramanathan V. Guha. 2000b. Resource Description Framework (RDF) Schema specification 1.0. World Wide Web Consortium, March. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>.
- [Bryan2003] Bryan, Martin. 2003. The HyTime Topic Maps (HyTM) Syntax 1.0. ISO/IEC JTC 1/SC34 N0391. <http://www.y12.doe.gov/sgml/sc34/document/0391.htm>.
- [Cannan and Otten1992] Cannan, Stephen and Gerard Otten. 1992. *SQL – the standard handbook*. McGraw-Hill Book Company, November.
- [Chandrasekaran1999] Chandrasekaran, Balakrishnan. 1999. What Are Ontologies, and Why do We Need Them? In *IEEE Intelligent Systems and their applications*, volume vl 9, n 1. IEEE, January.
- [Ciancarini et al.2003] Ciancarini, Paolo, Riccardo Gentilucci, Marco Pirruccio, Valentina Presutti, and Fabio Vitali. 2003. Metadata on the Web: On the integration of RDF and Topic Maps. In *Extreme Markup Languages 2003: Proceedings*. IDEAlliance. <http://www.idealliance.org/papers/extreme03/xslfo-pdf/2003/Presutti01/EML2003Presutti01.pdf>.
- [Clark and DeRose1999] Clark, James and Steve DeRose. 1999. XML Path Language (XPath) - Version 1.0. <http://www.w3.org/TR/xpath>, November.
- [Coakes2003] Coakes, Elayne. 2003. *Knowledge Management: Current Issues and Challenges*. Idea Group Publishing, April.
- [Connolly et al.2001] Connolly, Dan, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. 2001. A series of notes covering DAML+OIL as W3C technical reports. World Wide Web Consortium. <http://www.w3.org/TR/daml+oil-reference>.
- [Daconta, Obrst, and Smith2003] Daconta, Michael C., Leo J. Obrst, and Kevin T. Smith. 2003. *The Semantic Web : A Guide to the Future of XML, Web Services, and Knowledge Management*. John Wiley & Sons, May.
- [DARPA2001] DARPA. 2001. DAML. Darpa Agent Markup Language Program. <http://www.daml.org/>.
- [de Almeida2003] de Almeida, José João Dias. 2003. *Dicionários Dinâmicos Multi-Fonte*. Ph.D. thesis, Departamento de Informática, Escola de Engenharia, Universidade do Minho.
- [DeRose, Maler, and Orchard2001] DeRose, Steve, Eve Maler, and David Orchard. 2001. XML Linking Language (XLink) - Version 1.0. <http://www.w3.org/TR/xlink>, June.
- [DeRose et al.2002] DeRose, Steven, Ron Daniel Jr., Paul Grosso, Eve Maler, Jonathan Marsh, and Norman Walsh. 2002. XML Pointer Language (XPointer). <http://www.w3.org/TR/xptr>, August.
- [Dodds2001] Dodds, Leigh. 2001. Schematron: Validating XML Using XSLT. In *XSLT UK Conference*. Keble College, Oxford, England.

- [Duckett et al.2001] Duckett, Jon, Oliver Griffin, Stephen Mohr, Francis Norton, Nikola Ozu, Ian Stokes-Rees, Jeni Tennison, Kevin Williams, and Kurt Cagle. 2001. *Professional XML Schemas*. Wrox Press.
- [Elmasri and Navathe1994] Elmasri, Ramez and Shamkant B. Navathe. 1994. Fundamentals of Database Systems. In *Benjamin/Cummings Publishing*. Redwood City, USA.
- [Farquhar, Fikes, and Rice1996] Farquhar, Adam, Richard Fikes, and James Rice. 1996. The Ontolingua Server: A tool for Collaborative Ontology Construction. In *KWA96*. Banff, Canada.
- [Fellbaum1999] Fellbaum, Christiana. 1999. *WORDNET: an electronic lexical database and some of its applications*. Cambridge, MA: MIT Press.
- [Fensel2002] Fensel, Dieter. 2002. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, November.
- [Fensel et al.2001] Fensel, Dieter, Frank van Harmelen, Ian Horrocks, Deborah McGuinness, and Peter F. Patel-Schneider. 2001. Oil: An ontology infrastructure for the semantic web. In *IEEE Intelligent Systems*, volume 16(2), pages 38–44. IEEE.
- [Ferraiolo and Jackson2003] Ferraiolo, Jon and Dean Jackson. 2003. Scalable Vector Graphics (SVG) 1.1 Specification. World Wide Web Consortium, January. <http://www.w3.org/TR/SVG/>.
- [Fikes and McGuinness2001] Fikes, Richard and Deborah McGuinness. 2001. An Axiomatic Semantics for RDF, RDF-S, and DAML+OIL. World Wide Web Consortium. <http://www.w3.org/TR/daml+oil-axioms>.
- [Fokoué et al.2001] Fokoué, Achille, Bob Schloss, Shane Curcuru, and David Marston. 2001. XML Schema Quality Checker. AlphaWorks IBM. <http://www.alphaworks.ibm.com/tech/xmlsqc>.
- [Fowler and Scott2000] Fowler, Martin and Kendall Scott. 2000. *UML Essencial*. Bookman.
- [Freese2002] Freese, Eric. 2002. Using DAML+OIL as a Constraint Language for Topic Maps. In *XML Conference and Exposition 2002*. IDEAlliance. [http://www.idealliance.org/papers/xml02/dx\\_xml02/papers/05-03-03/05-03-03.html](http://www.idealliance.org/papers/xml02/dx_xml02/papers/05-03-03/05-03-03.html).
- [Fröhlich and Schmuck2004] Fröhlich, Christoph and Niko Schmuck. 2004. Semantag: Topic Map Generator. SourceForge. <http://developer.berlios.de/projects/semantag/>.
- [Frost1986] Frost, Richard. 1986. *Introduction to Knowledge Base Systems*. New York: MacMillan Publishing.
- [Garcia-Molina et al.1997] Garcia-Molina, Hector, Yannis Papakonstantinou, Dallen Quass, Anand Rajaraman, Yehoshua Sagiv, Jeffrey Ullman, Vasilis Vassalos, and Jennifer Widom. 1997. The TSIMMIS Approach to Mediation: Data Models and Languages. In

*Journal of Intelligent Information Systems*, volume 8(2), pages 117–132. Kluwer Academic Publishers.

- [Garshol2002a] Garshol, Lars Marius. 2002a. An RDF schema for topic maps. <http://psi.ontopia.net/rdf/>.
- [Garshol2002b] Garshol, Lars Marius. 2002b. Guide to the topic map standardization process. ISO/IEC JTC1 SC34 N0323. <http://www.y12.doe.gov/sgml/sc34/document/0323.htm>.
- [Garshol2002c] Garshol, Lars Marius. 2002c. LTM – The Linear Topic Map Notation. Ontopia. <http://www.ontopia.net/topicmaps/ltm.html>.
- [Garshol2003] Garshol, Lars Marius. 2003. Living with topic maps and RDF. In *Ontopia*. <http://www.ontopia.net/topicmaps/materials/tmrdf.html>.
- [Garshol2004a] Garshol, Lars Marius. 2004a. Metadata? Thesauri? Taxonomies? Topic Maps! Ontopia. <http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html>.
- [Garshol2004b] Garshol, Lars Marius. 2004b. The Ontopia Schema Language – Reference Specification. <http://www.ontopia.net/omnigator/docs/schema/spec.html>.
- [Garshol2004c] Garshol, Lars Marius. 2004c. The Ontopia Schema Language – Tutorial. <http://www.ontopia.net/omnigator/docs/schema/tutorial.html>.
- [Garshol2004d] Garshol, Lars Marius. 2004d. [tmcl-wg] Doubts in TMCL Requirements. Conversa e troca de e-mails (tmcl-wg@isotopicmaps.org). <http://www.isotopicmaps.org/pipermail/tmcl-wg/2004-February/000147.html>.
- [Garshol2004e] Garshol, Lars Marius. 2004e. tolog – A topic map query language. Ontopia. <http://www.ontopia.net/topicmaps/materials/tolog.html>.
- [Garshol and Barta2003] Garshol, Lars Marius and Robert Barta. 2003. TMQL requirements (1.2.0). ISO/IEC JTC1 SC34 N0448. <http://www.jtc1sc34.org/repository/0549.htm>.
- [Garshol and Barta2005] Garshol, Lars Marius and Robert Barta. 2005. Topic Map Constraint Language (TMCL). ISO/IEC JTC1/SC34. <http://www.isotopicmaps.org/tmql/spec.html>.
- [Garshol and Moore2005] Garshol, Lars Marius and Graham Moore. 2005. Topic Maps – Data Model. In *ISO/IEC JTC 1/SC34*. <http://www.isotopicmaps.org/sam/sam-model/>, January.
- [Garshol et al.2003] Garshol, Lars Marius, Graham Moore, Steven R. Newcomb, Michel Biezunski, and Martin Bryan. 2003. Topic Maps – Data Model, November.
- [Gennusa2004] Gennusa, Pamela. 2004. Ontopia’s Vizigator(tm) - Now you see it! In *XML 2004 Conference and Exposition*, Washington D.C., U.S.A. IDEAlliance. <http://www.idealliance.org/proceedings/xml04/papers/311/311.html>.

- [Gómez-Pérez and Corcho2002] Gómez-Pérez, Asunción and Oscar Corcho. 2002. Ontology Languages for the Semantic Web. In *IEEE Intelligent Systems*.
- [Godin, Missaoui, and Alaoui1995] Godin, Robert, Rokia Missaoui, and Hassan Alaoui. 1995. Incremental concept formation algorithms based on galois (concept) lattice. In *Computational Intelligence*, volume 11(2).
- [Goh1997] Goh, Cheng Hian. 1997. *Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources*. Ph.D. thesis, Phd, MIT.
- [Goldfarb et al.1997] Goldfarb, Charles, Steven R. Newcomb, W. Eliot Kimber, and Peter J. Newcomb. 1997. A Reader's Guide to the HyTime Standard. HyTime Users' Group. <http://www.hytime.org/papers/htguide.html>.
- [Grønmo2000] Grønmo, Geir Ove. 2000. Creating semantically valid topic maps. In *XML Europe 2000*. IDEAlliance. <http://www.gca.org/papers/xml europe2000/papers/s29-02.html>.
- [Gruber1993] Gruber, Thomas R. 1993. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands. Kluwer Academic Publishers.
- [Gruber1995] Gruber, Thomas R. 1995. How to Design an Ontology. In *Web page as part of the Ontolingua guided tour*. Accessed in April 2002.
- [Guarino1997] Guarino, Nicola. 1997. Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, pages 139–170, <http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/SCIE97.pdf>. Springer Verlag.
- [Guarino1998] Guarino, Nicola. 1998. Formal Ontology and Information Systems. In *Conference on Formal Ontology (FOIS98)*. <http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/FOIS98.pdf>.
- [Guarino and Giaretta1995] Guarino, Nicola and P. Giaretta. 1995. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N. Mars, editor, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pages 25–32. Ed. Amsterdam: ISO Press.
- [Hacker2000] Hacker, Scot. 2000. *MP3: The Definitive Guide*. O'Reilly.
- [Handschuh et al.2001] Handschuh, Staab, Alexander D. Maedche, Ljiljana Stojanovic, and Raphael Volz. 2001. KAON. <http://kaon.semanticWeb.org/kaon/white-paper.pdf>.
- [Harald Alvestrand2001] Harald Alvestrand. 2001. RFC 3066 - Tags for the Identification of Languages, January. IETF. <http://www.ietf.org/rfc/rfc3066.txt>.



- [Härder, Sauter, and Thomas1999] Härder, Theo, Günter Sauter, and Joachim Thomas. 1999. The intrinsic problems of structural heterogeneity and an approach to their solution. *The VLDB Journal*, 8(1):25–43.
- [Hasselbring2000] Hasselbring, Wilhelm. 2000. Information system integration. *Commun. ACM*, 43(6):32–38.
- [Heckel2001a] Heckel, Ronald. 2001a. Einsatzmöglichkeiten von Topic Maps zur flexiblen Navigation in elektronischen Dokumenten. Master's thesis, Lehrstuhl für Wirtschaftsinformatik, insbesondere Informationsmanagement Fakultät Wirtschaftswissenschaften, Technische Universität Dresden.
- [Heckel2001b] Heckel, Ronald. 2001b. Topic Map Designer tutorial. <http://www.topicmap-design.com/en/tutorial.htm>.
- [Hodges1997] Hodges, Wilfrid. 1997. *A Shorter Model Theory*. Cambridge University Press.
- [Holman2001] Holman, G. Ken. 2001. *Definitive XSLT and XPath*. Prentice Hall.
- [IEC2005] IEC. 2005. International Electrotechnical Commission - International Standards and Conformity Assessment. <http://www.iec.ch>.
- [IETF1998] IETF. 1998. Uniform Resource Identifiers (URI). Internet Engineering Task Force, August. <http://www.ietf.org/rfc/rfc2396.txt>.
- [ISO1986] ISO. 1986. ISO 2788:1986 – Documentation – Guidelines for the establishment and development of monolingual thesauri.
- [ISO1997] ISO. 1997. ISO 3166:1997 – Code for the representation of name of countries and their subdivisions. part 1 – 3.
- [ISO1998] ISO. 1998. ISO 639:1997 Code for the representation of name of languages.
- [ISO2005] ISO. 2005. International Organization for Standardization. <http://www.iso.org>.
- [Jacinto et al.2002a] Jacinto, Marta H., Giovanni R. Librelotto, José C. Ramalho, and Pedro R. Henriques. 2002a. Constraint Specification Languages: comparing XCSL, Schema-tron and XML-Schemas. In *XML Europe 2002*, Barcelona, Spain.
- [Jacinto et al.2002b] Jacinto, Marta H., Giovanni R. Librelotto, José C. Ramalho, and Pedro R. Henriques. 2002b. XCSL: XML Constraint Specification Language. In *XXVIII Conferencia Latinoamericana de Informtica*, Montevideo, Uruguay.
- [Jacinto et al.2003] Jacinto, Marta H., Giovanni R. Librelotto, José C. Ramalho, and Pedro R. Henriques. 2003. XCSL: XML Constraint Specification Language. In *CLEI Electronic Journal*.
- [Johannesen2003] Johannesen, Alexander. 2003. Compact Simplified XTM: CSXTM 0.3. <http://www.shelter.nu/csxtm.html>.

- [Johannesen2004] Johannesen, Alexander. 2004. xSiteable, March. <http://www.shelter.nu/xsiteable/>.
- [Karp, Chaudhri, and Thomere1999] Karp, Peter D., Vinay K. Chaudhri, and Jerome Thomere. 1999. Xol: an xml-based ontology exchange language. <http://www.ai.r.sri.com/pkarp/xol>.
- [Kashyap and Sheth1996] Kashyap, Vipul and Amit P. Sheth. 1996. Schematic and semantic similarities between database objects: A context-based approach. In *The International Journal on Very Large Data Bases*, volume 5(4), pages 276–304. VLDB Journal.
- [Kay1979] Kay, Martin. 1979. Functional grammar. *5th Annual Meeting of the Berkeley Linguistic Society*.
- [Kifer, Lausen, and Wu1995] Kifer, Michael, Georg Lausen, and James Wu. 1995. Logical Foundations of Object-Oriented and Frame-Based Languages. In *Journal of the ACM*.
- [Kim and Seo1991] Kim, Won and Jungyun Seo. 1991. Classifying schematic and data heterogeneity in multidatabase systems. In *IEEE Computer*, volume 24(12), pages 12–18. IEEE.
- [Lacher and Decker2001] Lacher, Martin S. and Stefan Decker. 2001. On the Integration of Topic Maps and RDF Data. In *Extreme Markup Languages 2001: Proceedings*. IDEAlliance. <http://www.idealliance.org/papers/extreme03/html/2001/Lacher01/EML2001Lacher01.html>.
- [Lamport and Paulson1999] Lamport, Leslie and Lawrence C. Paulson. 1999. Should your specification language be typed. *ACM Trans. Program. Lang. Syst.*, 21(3):502–526.
- [Lassila and Swick1999] Lassila, Ora and Ralph R. Swick. 1999. Resource Description Framework (RDF) Model and Syntax Specification. World Wide Web Consortium, February. <http://www.w3.org/TR/REC-rdf-syntax>.
- [Lenat and Guha1990] Lenat, Douglas B. and Ramanathan V. Guha. 1990. Building large knowledge-based systems. Representation and Inference in the yc Project. In *Addison-Wesley, Readings, Massachusetts*.
- [Librelotto, Ramalho, and Henriques2003a] Librelotto, Giovanni R., José C. Ramalho, and Pedro R. Henriques. 2003a. ADRIAN – a platform for e-learning content production. In *Second International Conference on Multimedia and Information & Communication Technologies in Education*.
- [Librelotto, Ramalho, and Henriques2003b] Librelotto, Giovanni R., José C. Ramalho, and Pedro R. Henriques. 2003b. Geração automática de interfaces Web para Sistemas de Informação: Metamorphosis. In *CoopMedia*, Porto, Portugal.
- [Librelotto, Ramalho, and Henriques2003c] Librelotto, Giovanni R., José C. Ramalho, and Pedro R. Henriques. 2003c. Ontology driven Websites with Topic Maps. In *The International Conference on Web Engineering*, Oviedo, Spain.

- [Librelotto, Ramalho, and Henriques2003d] Librelotto, Giovani R., José C. Ramalho, and Pedro R. Henriques. 2003d. TM-Builder: Um Construtor de Ontologias baseado em Topic Maps. In *XXIX Conferencia Latinoamericana de Informática*, La Paz, Bolívia.
- [Librelotto, Ramalho, and Henriques2004a] Librelotto, Giovani R., José C. Ramalho, and Pedro R. Henriques. 2004a. ADRIAN E-Learning Content Production (creating online exams). In *VIII International Conference on Electronic Publishing*, Brasília, Brasil.
- [Librelotto, Ramalho, and Henriques2004b] Librelotto, Giovani Rubert, José Carlos Ramalho, and Pedro Rangel Henriques. 2004b. Extração de Topic Maps no Oveia: Especificação e Processamento. In Mauricio Solar, David Fernández-Baca, and Ernesto Cuadros-Vargas, editors, *30ma Conferencia Latinoamericana de Informática (CLEI2004)*, pages 451–460. Sociedad Peruana de Computación, September. ISBN 9972-9876-2-0.
- [Librelotto, Ramalho, and Henriques2004c] Librelotto, Giovani Rubert, José Carlos Ramalho, and Pedro Rangel Henriques. 2004c. Ulisses - A Topic Maps driven Conceptual Navigator. <http://www.di.uminho.pt/~gepl/metamorphosis/ulisses.html>.
- [Librelotto et al.2004] Librelotto, Giovani Rubert, Weber Souza, José Carlos Ramalho, and Pedro Rangel Henriques. 2004. Using the Ontology Paradigm to Integrate Information Systems. In *International Conference on Knowledge Engineering and Decision Support*, pages 497–504, Porto, Portugal.
- [Liu and Cleary1996] Liu, Mengchi and John Cleary. 1996. Deductive Databases – Where to Now? In *Journal of Intelligent Information Systems*. <http://www.scs.carleton.ca/~mengchi/papers/ddb-JIIS.ps>.
- [Lomet and Widom1995] Lomet, David B. and Jennifer Widom, editors. 1995. *Special Issue on Materialized Views and Data Warehousing*, IEEE Data Engineering Bulletin 18(2), June.
- [Luke and Helfin2000] Luke, Sean and Jeff Helfin. 2000. Shoe 1.01 proposed specification. SHOE Project. <http://www.csumd.edu/projects/plus/SHOE/spec.html>.
- [Lutz and Ascher2003] Lutz, Mark and David Ascher. 2003. *Learning Python*. O'Reilly & Associates.
- [Mack and Yael Ravin2001] Mack, Robert and Roy J. Byrd Yael Ravin. 2001. Knowledge portals and the emerging digital knowledge workplace. *IBM System Journal*, 40(4):925–955.
- [March and Olsen1976] March, James and Johan Olsen. 1976. *Ambiguity and Choice in Organizations*. Bergen, Norway:Universitetsforlaget.
- [Mason and Desautels2002] Mason, James David and Sara Hafele Desautels. 2002. National Body Comments on SC 34 N 331 Montreal meeting of WG3 and Documents SC 34 N 327 Notes on Issues to be decided on scope, SC 34 N 328 The XML Topic Maps (XTM) Syntax 1.0, and SC 34 N 329 The Standard Application Model for Topic Maps. ISO/IEC JTC 1/SC34 N0338. <http://www.y12.doe.gov/sgml/sc34/document/0338.htm>.

- [McGregor1991] McGregor, R. 1991. Inside the LOOM classifier. In *SIGART bulletin*.
- [McGuinness and van Harmelen2004] McGuinness, Deborah L. and Frank van Harmelen. 2004. Web Ontology Language (OWL) – Overview. <http://www.w3.org/TR/owl-features/>, February.
- [Microsoft2003] Microsoft. 2003. Microsoft SQL Server XML Extensions (SQLXML) 3.0 SP1. <http://download.microsoft.com/download/SQLSVR2000/SP/3.0SP1/WIN98MeXP/EN-US/sqlxml.msi>.
- [Mitra and Wiederhold2001] Mitra, Prasenjit and Gio Wiederhold. 2001. An algebra for semantic interoperability of information sources. In *BIBE '01: Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering*, page 174, Washington, DC, USA. IEEE Computer Society.
- [Moore, Nishikawa, and Bogachev2004] Moore, Graham, Mary Nishikawa, and Dmitry Bogachev. 2004. Topic Map Constraint Language. ISO/IEC JTC 1/SC 34. <http://www.jtc1sc34.org/repository/0549.htm>.
- [Motta1999] Motta, Enrico. 1999. Reusable Components for Knowledge Modeling. In *IOS Press*.
- [Mylopoulos et al.1990] Mylopoulos, John, Alex Borgida, Matthias Jarke, and Manolis Koubarakis. 1990. Telos: representing knowledge about information systems. *ACM Trans. Inf. Syst.*, 8(4):325–362.
- [Newcomb2003] Newcomb, Steven R. 2003. A Semantic Integration Methodology. In *Extreme Markup Languages 2003: Proceedings*. IDEAlliance. <http://www.idealliance.org/papers/extreme03/html/2003/Newcomb01/EML2003Newcomb01.html>.
- [Newcomb, Biezunski, and Bryan2003] Newcomb, Steven R., Michel Biezunski, and Martin Bryan. 2003. The HyTime Topic Maps (HyTM) Syntax 1.0. ISO/IEC JTC 1/SC34 N0391. <http://www.jtc1sc34.org/repository/0391.htm>.
- [Nishikawa2003] Nishikawa, Mary. 2003. A metadata network for bridging people and places. In *Extreme Markup Languages 2003: Proceedings*. IDEAlliance. <http://www.idealliance.org/papers/extreme03/html/2003/Nishikawa01/EML2003Nishikawa01.html>.
- [Nishikawa and Moore2003a] Nishikawa, Mary and Graham Moore. 2003a. Requirements for a Topic Map Constraint Language JTC 1 NP Number. ISO/IEC 19756. ISO/IEC JTC 1/SC34 N0405. <http://www.y12.doe.gov/sgml/sc34/document/0405.htm>.
- [Nishikawa and Moore2003b] Nishikawa, Mary and Graham Moore. 2003b. Topic Map Constraint Language (TMCL) Requirements and Use Cases. ISO/IEC JTC 1/SC34 N0405rev. <http://www.isotopicmaps.org/tmcl/requirements.html>.
- [Nishikawa, Moore, and Bogachev2004] Nishikawa, Mary, Graham Moore, and Dmitry Bogachev. 2004. Topic Map Constraint Language (TMCL) Requirements and Use Cases. ISO/IEC JTC 1/SC34 N0548. <http://www.jtc1sc34.org/repository/0548.htm>.

- [Niskanen2000] Niskanen, Pekka. 2000. *Inside WAP: Programming Applications with WML and WMLScript*. Addison-Wesley Professional.
- [Novak1977] Novak, Joseph D. 1977. *A Theory of education*. Ithaca, N.Y., Cornell University Press.
- [Noy and McGuinness2001] Noy, Natalya F. and Deborah L. McGuinness. 2001. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University, Stanford. <http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noymcguinness.html>.
- [OASIS2003a] OASIS. 2003a. Draft Requirements Document on Topic Maps Published Subjects. Cover Pages – OASIS. <http://xml.coverpages.org/ni2003-03-10-b.html>.
- [OASIS2003b] OASIS. 2003b. OASIS Topic Maps Published Subjects TC. OASIS. [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=tm-pubsubj](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tm-pubsubj).
- [Ogievetsky2000] Ogievetsky, Nikita. 2000. XSLT stylesheets for converting ISO 13250 Topic Map documents into XTM 1.0 syntax. <http://www.cogx.com/xslt4tm2xtm.html>.
- [OMG2003] OMG. 2003. OMG Unified Modeling Language Specification. Object Management Group. <http://www.omg.org/docs/formal/03-03-01.pdf>.
- [Ontopia2002a] Ontopia. 2002a. Ontopia Navigator Framework. <http://www.ontopia.net/solutions/navigator.html>.
- [Ontopia2002b] Ontopia. 2002b. The Ontopia Omnigator. <http://www.ontopia.net/omnigator/>.
- [Ontopia2004] Ontopia. 2004. The Ontopia Knowledge Suite, February. <http://www.ontopia.net/solutions/products.html>.
- [Oracle2002] Oracle. 2002. Oracle XML SQL Utility - XSU. [http://technet.oracle.com/tech/xml/oracle\\_xsu/content.html](http://technet.oracle.com/tech/xml/oracle_xsu/content.html).
- [Pacheco et al.1999] Pacheco, José Augusto, Maria Palmira Alvez, Maria Assunção Flores, João M. Paraskeva, José Carlos Morgado, Ana Maria Silva, and Isabel Carvalho Serra. 1999. *Componentes do Processo de Desenvolvimento do Currículo*. Coleção Minho Universitária, Livraria Minho.
- [Park and Hunting2003] Park, Jack and Sam Hunting. 2003. *XML Topic Maps: Creating and Using Topic Maps for the Web*, volume ISBN 0-201-74960-2. Addison Wesley.
- [Patel-Schneider et al.2004] Patel-Schneider, Peter F., Patrick Hayes, Ian Horrocks, and Frank van Harmelen. 2004. Web Ontology Language (OWL) – Abstract Syntax and Semantics. <http://www.w3.org/TR/2004/REC-owl-semantic-20040210/>, February.
- [Pepper1999] Pepper, Steve. 1999. Euler, Revolution, and Topic Maps. In *XML Europe*. <http://www.ontopia.net/topicmaps/materials/euler.pdf>.

- [Pepper2000] Pepper, Steve. 2000. The TAO of Topic Maps - finding the way in the age of infoglut. Ontopia. <http://www.ontopia.net/topicmaps/materials/tao.html>.
- [Pepper2002] Pepper, Steve. 2002. Ten Thesis on Topic Maps and RDF. <http://www.ontopia.net/topicmaps/materials/rdf.html>, August.
- [Pepper2003] Pepper, Steve. 2003. Published Subjects: Introduction and Basic Requirements. OASIS. OASIS Technical Committee Working Draft. Produced in the Topic Maps Published Subjects Technical Committee.
- [Pepper and Moore2001a] Pepper, Steve and Graham Moore. 2001a. XML Topic Maps (XTM) 1.0. TopicMaps.Org Specification, August. <http://www.topicmaps.org/xtm/1.0/>.
- [Pepper and Moore2001b] Pepper, Steve and Graham Moore. 2001b. XML Topic Maps (XTM) 1.0 - Annex D: XTM 1.0 Document Type Declaration (Normative). TopicMaps.Org Specification, August. <http://www.topicmaps.org/xtm/1.0/#dtd>.
- [Pepper and Schwab2003] Pepper, Steve and Sylvia Schwab. 2003. Curing the Web's Identity Crisis: Subject Indicators for RDF. In *XML Europe 2003*. <http://www.ontopia.net/topicmaps/materials/identitycrisis.html>.
- [Powers2003] Powers, Shelley. 2003. *Practical RDF*. O'Reilly & Associates, 1st edition edition, July.
- [Prescott, Harley, and Harley1996] Prescott, Lansing M., John P. Harley, and John Harley. 1996. *Microbiology*. Wm. C. Brown Pub. Dubuque, Iowa, third edition edition. pp. 390-414.
- [Protégé2005] Protégé. 2005. The protégé ontology editor and knowledge acquisition system. <http://protege.stanford.edu/>.
- [Ramalho2000] Ramalho, José Carlos Leite. 2000. *Anotação Estrutural de Documentos e sua Semântica*. Ph.D. thesis.
- [Rath2003] Rath, H. Holger. 2003. White Paper: The Topic Maps Handbook. Empolis.
- [Rath and Pepper1999] Rath, Hans Holger and Steve Pepper. 1999. Topic Maps: Introduction and Allegro. In *Markup Technologies 99 Proceedings*, December.
- [Ricker2000] Ricker, Jeffrey. 2000. *Managing Metadata With XML and RDF: Improving Workflow for Web Applications*. John Wiley & Sons, December.
- [Schreiber et al.1994] Schreiber, Guus, Bob Wielinga, Hans Akkermans, Walter van de Velde, and Anjo Anjewierden. 1994. CML: The CommonKADS Conceptual Modeling Language. In Steels et al., editor, *A Future of Knowledge Acquisition, Proc. 8th European Knowledge Acquisition Workshop (EKAW94)*, Hoegaarden. Lecture Notes in Artificial Intelligence 867, Springer-Verlag.
- [Shannon1974] Shannon, Claude E. 1974. A Mathematical Theory of Communication. In *Key Papers in the Development of Information Theory*. New York: IEEE.

- [Shapiro2002] Shapiro, Alexander. 2002. TG Development. <http://touchgraph.sourceforge.net/>.
- [Sigel2000] Sigel, Alexander. 2000. Towards knowledge organization with topic maps. In *XML Europe 2000 Conference*, Le Palais des Congrès de Paris, Paris, France. GCA.
- [Smith, Welty, and McGuinness2004] Smith, Michael K., Chris Welty, and Deborah McGuinness. 2004. Web Ontology Language (OWL) – Guide Version 1.0. <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>, February.
- [SourceForge2003a] SourceForge. 2003a. Nexist. <http://nexist.sourceforge.net/>.
- [SourceForge2003b] SourceForge. 2003b. Topic Maps for Java - TM4J. <http://tm4j.org/>.
- [SourceForge2005] SourceForge. 2005. TMNav. <http://tm4j.org/tmnav.html>.
- [Sowa2000] Sowa, John F. 2000. *Knowledge Representation: logical, philosophical and computational foundations*. Brooks/Cole.
- [Staab et al.2000] Staab, Steffen, Michael Erdmann, Alexander Maedche, and Stefan Decker. 2000. An extensible approach for modeling ontologies in RDF(S). ECDL 2000 Workshop on the Semantic Web. <http://www.y12.doe.gov/sgml/sc34/document/0391.htm>.
- [Su and Ilebrekke2002] Su, Xiaomeng and Lars Ilebrekke. 2002. A Comparative Study of Ontology Languages and Tools. In *CAISE 02*. Toronto, Canada.
- [Swartout and Tate1999] Swartout, William and Austin Tate. 1999. Ontologies. In *IEEE Intelligent Systems and their applications*, volume vl 14, n 1. IEEE, January.
- [Teorey1999] Teorey, Toby. 1999. *Database Modeling and Design*. San Francisco, CA: Morgan Kaufman, 3rd edition.
- [Thompson2003] Thompson, Henry. 2003. The Extensible Stylesheet Language (XSL) Family. <http://www.w3.org/Style/XSL/>.
- [Ullman and Zaniolo1990] Ullman, Jeffrey D. and Carlo Zaniolo. 1990. Deductive Databases: Achievements and Future Directions. In *SIGMOD Record 19(4)*, 75–82. <http://www.cs.ucla.edu/~zaniolo/papers/sigrecord90.pdf>.
- [UNESCO1995] UNESCO. 1995. UNESCO Thesaurus. United Nations Educational, Scientific and Cultural Organization. <http://www.ulcc.ac.uk/unesco/>.
- [Uschold and Gruninger1996] Uschold, Mike and Michael Gruninger. 1996. Ontologies: Principles, methods and applications. In *Knowledge Engineering Review*, volume 11(2), pages 93–155. Cambridge University Press.
- [van Dalen1994] van Dalen, Dirk. 1994. *Logic and Structure*. ISBN: 3-540-57839-0. Springer, 3rd augmented edition, (corrected 2nd printing 1997) edition.

- [Vieira2002] Vieira, André Accioly. 2002. Extração de Ontologias a partir de Esquemas Relacionais. Instituto Militar de Engenharia. Mestrado em Sistemas e Computação.
- [Vieira, Tanaka, and Moura2002] Vieira, André Accioly, Astério Kiyoshi Tanaka, and Ana Maria Carvalho Moura. 2002. OntoExtract – Ferramenta para Extração de Ontologias a partir de Bancos de Dados Relacionais. *WTDBD/SBBD/2002*.
- [Volz2003] Volz, Raphael. 2003. KAON REVERSE. <http://kaon.semanticweb.org/alphaworld/reverse/view>.
- [W3C2005] W3C. 2005. World Wide Web Consortium. <http://www.w3.org>.
- [Wache et al.2001] Wache, Holger, T. Vögele, Ubbo Visser, Heiner Stuckenschmidt, Gerhard Schuster, Holger Neumann, and Sebastian Hübner. 2001. Ontology-based integration of information — a survey of existing approaches. In H. Stuckenschmidt, editor, *IJCAI-01 Workshop: Ontologies and Information Sharing*, pages 108–117.
- [Walsh and Muellner1999] Walsh, Norman and Leonard Muellner. 1999. *Docbook: The Definitive Guide*. O'Reilly & Associates.
- [Web3D2004] Web3D. 2004. Web 3D Consortium – Open Standards for Real-Time 3D Communication, March. <http://www.web3d.org/>.
- [Williams et al.2000] Williams, Kevin, Michael Brundage, Patrick Dengler, Jeff Gabriel, Andy Hoskinson, Michael Kay, Thomas Maxwell, Marcelo Ochoa, Johnny PaPa, and Mohan Vanmane. 2000. *Professional XML Databases*. Wrox Press.
- [Wrightson2001] Wrightson, Ann. 2001. Topic Maps and Knowledge Representation. Ontopia, February. <http://www.ontopia.net/topicmaps/materials/kr-tm.html>.
- [Yoshikawa, Amagasa, and Uemura2001] Yoshikawa, Masatoshi, Toshiyuki Amagasa, and Shunsuke Uemura. 2001. XRel: A path-based approach to storage and retrieval of XML documents using relational databases. Nata Institute of Science and Technology.



## Apêndice A

# Regras para a Validação de Especificações XTche

O presente apêndice apresenta as regras que devem ser obedecidas por uma especificação XTche para ser considerada válida. Tais regras são implementadas pelo processador *XTche-Validator*, apresentado na Subseção 9.5.2.

O processador *XTche-Validator* implementa o conjunto de regras abaixo, as quais divididas em 3 sub-conjuntos: **restrições em esquemas**, **restrições contextuais** e **restrições de existência**.

### 1. Regras para Restrições de Esquema:

- (a) o primeiro nível do elemento `<schema-constraints>` deve ter elementos que possuam somente um destes atributos: `associationType`, `occurrenceType`, `topicType`, ou `baseNameType`;
- (b) elementos com o atributo `associationType` devem ter sub-elementos com somente um destes atributos: `associationScope`, `associationRole-Exclusive`, `associationRole`, `associationPlayer` ou `associationPlayer-Exclusive`;
- (c) elementos com o atributo `associationType` podem ter o sub-elemento `<xs:any>`. Seu atributo `namespace` deve possuir o valor `##any`;
- (d) elementos que possuam atributos com o sufixo `Exclusive` não contém sub-elementos;
- (e) elementos com os atributos `associationScope` ou `associationPlayer` não devem conter nenhum sub-elemento;
- (f) elementos com o atributo `associationRole` podem ter sub-elementos com o atributo `associationPlayer` e `topicType`;
- (g) elementos com o atributo `associationRole` podem possuir atributos `minOccurs` e `maxOccurs`;
- (h) elementos com o atributo `associationRole` podem ter o sub-elemento `<xs:any>`. Seu atributo `namespace` deve possuir o valor `##any`;

- (i) elementos com o atributo `associationPlayer` podem possuir atributos `minOccurs` e `maxOccurs`;
- (j) elementos com o atributo `topicType` devem ter sub-elementos com somente um destes atributos: `baseName`, `baseNameScope`, `occurrenceType`, `occurrenceScope`, `subjectIndicator` ou `topic`.
- (k) elementos com os atributos `baseNameScope` ou `occurrenceScope` não contém sub-elementos;
- (l) elementos com os atributos `occurrenceType` ou `occurrenceScope` podem ter atributos `minOccurs` e `maxOccurs`;
- (m) elementos com o atributo `topicType` podem ter o sub-elemento `<xs:any>`;
- (n) elementos com o atributo `occurrenceType` podem ter sub-elementos com o atributo `occurrenceScope`.
- (o) os elementos `tms:baseName`, `tms:occurrence` e `tms:subjectIndicator` devem ser sub-elementos de um elemento que necessariamente seja qualificado com o atributo `topicType` ou o atributo `topic`.
- (p) os elementos `tms:baseName`, `tms:occurrence` podem possuir atributos `minOccurs` e `maxOccurs`;

## 2. Regras para Restrições Contextuais:

- (a) atributos com o sufixo `Forbidden` devem somente ser encontrados em sub-elementos filhos do elemento `<contextual-conditions>`;
- (b) atributos com o sufixo `Exclusive` devem ser únicos em sub-elementos de `<contextual-conditions>`;
- (c) um elemento pode ter mais de um atributo com o sufixo `Forbidden`, mas todos os seus atributos devem possuir tal sufixo.

## 3. Regras para Restrições de Existência:

- (a) o primeiro nível do elemento `<existence-constraints>` deve ter elementos que possuam somente um destes atributos: `associationType`, `topicType` ou `topic`;
- (b) nenhum elemento da sub-árvore de `<existence-constraints>` pode ter atributos com o sufixo `Forbidden` e `Exclusive`;
- (c) o elemento `<xs:any>` não pode ser encontrado nesta sub-árvore;
- (d) nenhum elemento desta sub-árvore pode ter atributos `minOccurs` e `maxOccurs`;
- (e) as regras 1(e), 1(j), 1(k), 1(n), 1(o) e 1(p) são válidas no contexto de restrições de existência;
- (f) elementos com o atributo `associationType` devem ter sub-elementos com somente um destes atributos: `associationScope`, `associationRole` ou `associationPlayer`;
- (g) elementos com o atributo `associationRole` podem ter sub-elementos com o atributo `associationPlayer` e `topic`.

## Apêndice B

# Comparando XTche, AsTMa! e OSL

O presente apêndice apresenta as linguagens que permitem verificações semânticas sobre Topic Maps, nomeadamente AsTMa! (*Asymptotic Topic Maps*) (Barta, 2003a) e OSL (*Ontopia Schema Language*) (Garshol, 2004c). Além de uma descrição da sintaxe e o processamento de cada linguagem, realiza-se um comparativo com XTche, ressaltando os seus pontos positivos e negativos na comparação com as demais.

### B.1 A linguagem de restrições AsTMa!

O objectivo desta linguagem de restrição é especificar restrições para topic maps. AsTMa! é membro da família de linguagens AsTM\* (Barta, 2003b) da qual também faz parte a linguagem AsTMa= (Barta, 2004), apresentada na Subsecção 3.4.2. A sintaxe de AsTMa! estende a sintaxe de AsTMa=, através de quantificadores e operadores extras adicionados a essa linguagem. As declarações de tópicos e associações na sintaxe de AsTMa= formam a base dos *maplet patterns*, o que será descrito na próxima subsecção.

#### B.1.1 Maplet Patterns

*Maplet patterns* são expressões na linguagem AsTMa! que determinam a especificação de padrões que devem ser seguidos pelo topic map. Eles possuem a mesma sintaxe que os tópicos e as associações têm em AsTMa=. Abaixo tem-se um exemplo de um *maplet pattern* em AsTMa= que identifica um tópico que possui um nome:

```
1 | * (pessoa)
2 | bn : /name/i
```

No caso acima, o identificador do tópico é substituído pelo curinga (*wildcard*) “\*” e o valor do nome é substituído pela expressão regular `/name/i`. Durante o processo de validação,

o mecanismo de restrições tentará encontrar um tópico ou associação dentro do topic map que está de acordo com o padrão. Por exemplo, a seguinte declaração de tópico em AsTMa= está de acordo com o *maplet pattern* acima:

```
1 | prh
2 | bn : Pedro Rangel Henriques
```

Variáveis também podem ser utilizadas dentro de um *maplet pattern*. Essas variáveis estão identificadas pelo caracter “\$” no início de sua declaração. A declaração do tópico acima está de acordo com o padrão abaixo, onde a variável \$id está restrita ao tópico com o identificador *prh*, e a variável \$basename está restrita ao valor *Pedro Rangel Henriques*.

```
1 | $id (pessoa)
2 | bn : $basename
```

Em AsTMa!, as restrições de existência são denotadas pela palavra-chave *exists*. Se *p* é um *maplet pattern*, então “*exists p*” define uma *restrição de existência*. Um topic map obedece a restrição de existência se existe ao menos um tópico ou associação que esteja de acordo com *p* no topic map. Um exemplo deste tipo de restrição, indicando que deve existir uma associação do tipo *orientar* onde o papel de actuação *orientador* é desempenhado pelo tópico *prh* e o papel *orientando*, pelo tópico *giovani*, está abaixo codificado:

```
1 | exists [ (orientar)
2 |         orientador : prh
3 |         orientando : giovani [
```

A linguagem AsTMa! fornece a opção de seleccionar um sub-mapa específico para validá-lo de acordo com uma restrição em particular através da palavra-chave *forall*, a qual funciona como o selector de sub-mapas; assim, as restrições de existência podem ser utilizadas para validar o sub-mapa de acordo com a regra desejada.

Se *p* é um *maplet pattern* e *c* é uma restrição, então “*forall p ⇒ c*” é uma *restrição condicional*. Para exemplificar, considere o seguinte exemplo:

```
1 | forall [ $p (pessoa) ]
2 | => exists [ (orientar)
3 |             * : $p ]
```

Essa regra define que todo tópico do tipo *pessoa* deve participar de uma associação do tipo *orientar*, independente do papel de actuação desempenhado por cada tópico deste tipo.

O processo de concordância de padrões (*pattern matching*) em AsTMa! pode ser estendido com duas diferentes regras, as quais cobrem os *maplet patterns* abertos e fechados.

Um *maplet pattern* fechado é denotado pelo uso de “]” em torno do padrão. Se *p* é um *maplet pattern*, então “[*p*]” é um *closed maplet pattern*. Quaisquer características adicionais no tópico ou membros adicionais na associação determinará que haverá uma

falha na concordância. Um exemplo de *maplet pattern* fechado, indicando que todos os tópicos do tipo *pessoa* devem possuir somente um nome e nenhuma outra característica, encontra-se a seguir:

```

1 forall [ $p (pessoa) ]
2   => exists [ $p
3             bn : * ]

```

Por sua vez, os *maplet patterns* abertos são denotados com o uso de “[ ]” ao redor do padrão. Se *p* é um *maplet pattern*, então “[*p*]” é um *open maplet pattern*, como o abaixo:

```

1 forall [ $p (pessoa) ]
2   => exists [ $p
3             bn : * ]

```

O *maplet pattern* acima define que todo tópico do tipo *pessoa* deve possuir um nome. Além disso, os tópicos deste tipo podem possuir quaisquer outras características, por este ser um *maplet pattern* aberto.

## B.2 A linguagem de restrições OSL

*Ontopia Schema Language* (OSL) (Garshol, 2004c) é uma linguagem que permite a definição de esquemas de topic maps, isto é, um conjunto de classes de tópicos e associações que um topic map deve seguir. Essas classes restringem as estruturas de suas instâncias e controlam a forma como a informação vai se apresentar num topic map que segue o esquema.

A validação de um topic map de acordo com um esquema é realizada ao encontrar as instâncias de cada classe de tópico ou associação, verificando se tal instância segue a definição da classe.

A linguagem OSL foi projectada para possuir um pequeno número de características e um poder de expressão mínimo (Garshol, 2004b). A Ontopia<sup>1</sup>, quem a desenvolve, pretende estender seu conjunto de características de acordo com a demanda oriunda dos utilizadores, e eventualmente, substituí-la por TMCL. OSL não implementa, portanto, todos os requisitos de TMCL.

Os principais objectivos de OSL são (Garshol, 2004b):

- validação da estrutura das instâncias de topic maps;
- documentação da estrutura de topic maps;
- fornecer um guia para a fácil edição de topic maps.

OSL utiliza uma sintaxe XML definida por um DTD. O exemplo a seguir representa uma regra onde todos os tópicos do tipo *pessoa* devem possuir somente um nome.

<sup>1</sup><http://www.ontopia.com>

```

1 <tm-schema>
2   <topic>
3     <instanceOf>
4       <internalTopicRef href="#pessoa"/>
5     </instanceOf>
6     <baseName min="1" max="1">
7       <scope></scope>
8     </baseName>
9   </topic>
10 </tm-schema>

```

O elemento `internalTopicRef` (linha 4) define qual é o tipo dos tópicos que terão que cumprir essa restrição. O elemento `baseName` (linha 6) possui dois atributos que determinam a sua cardinalidade (`@min` e `@max`) e um sub-elemento `scope`, sem conteúdo (linha 7), indicando que este nome não deve estar inserido em contexto algum.

## B.3 Comparando XTche, AsTMa! e OSL

Essa subsecção apresenta uma série de situações que devem ser validadas por uma linguagem de especificação de restrições em **Topic Maps**. Em cada situação, é apresentado como cada uma das linguagens apresentadas anteriormente permitem definir o conjunto de regras para a efectiva validação.

### B.3.1 Condições estruturais rígidas

A primeira situação que se apresenta é o caso onde tem-se que definir que toda associação do tipo A deve possuir apenas os papéis de actuação P1 e P2; além desses, nenhum outro papel deve ser encontrado em qualquer associação deste tipo.

Um exemplo deste tipo de restrição é: as associações do tipo *orientar* devem conter somente os papéis de actuação *orientador* e *orientando*. Ao confrontar esta regra com o segmento de topic map abaixo, deve ser detectado um erro, pois o papel de actuação *universidade* (linha 19) não era suposto estar presente neste tipo de associação.

```

1 <association>
2   <instanceOf>
3     <topicRef xlink:href="#orientar"/>
4   </instanceOf>
5   <member>
6     <roleSpec>
7       <topicRef xlink:href="#orientador"/>
8     </roleSpec>
9     <topicRef xlink:href="#prh"/>
10  </member>
11  <member>
12    <roleSpec>
13      <topicRef xlink:href="#orientando"/>
14    </roleSpec>
15    <topicRef xlink:href="#giovani"/>
16  </member>
17 </member>

```

```

18         <roleSpec>
19             <topicRef xlink:href="#universidade"/>
20         </roleSpec>
21         <topicRef xlink:href="#uminho"/>
22     </member>
23 </association>

```

A definição deste tipo de regra em XTche encontra-se na Figura B.1. Nessa regra percebe-se que o tipo de associação *orientar* (elemento com o atributo `@xtche:associationType`) possui dois papéis de actuação: *orientando* e *orientador* (elementos com o atributo `@xtche:associationRole`). Qualquer outro papel encontrado em uma associação deste tipo, como o caso do papel *universidade*, faz com que o TM-Validator informe uma mensagem de erro, invalidando o topic map.

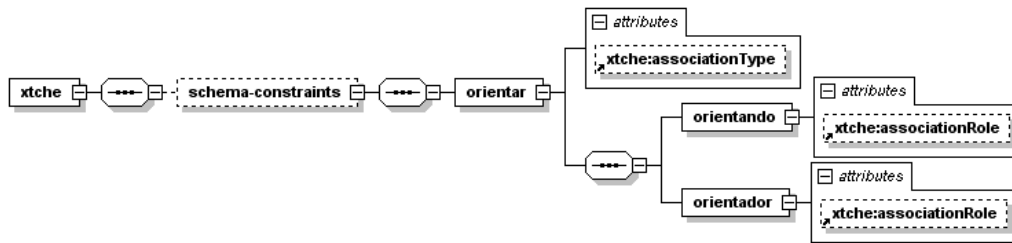


Figura B.1: Regra XTche para a associação *orientar* referente a um *closed maplet pattern*

Em AsTMa!, uma regra para resolver esta restrição deve ser escrita num *closed maplet pattern*, como o exemplo encontra-se a seguir:

```

1 forall $a [ (orientar) ]
2     => exists $a [ (orientar)
3         orientador : *
4         orientando : * [

```

Esse *maplet pattern* fechado não será satisfeito pela associação encontrada no topic map anterior, pois apesar de haver uma concordância entre o tipo da associação e dois de seus papéis, o papel de actuação *universidade* não faz parte deste padrão.

A linguagem OSL permite a especificação deste tipo de restrição no elemento `association`, definindo o tipo da associação no elemento `instanceOf` e os papéis de actuação possíveis nos elementos `role`. O exemplo abaixo apresenta o código OSL para a associação *orientar*:

```

1 <association>
2   <instanceOf>
3     <internalTopicRef href="#orientar"/>
4   </instanceOf>
5   <role min="1" max="1">
6     <instanceOf>
7       <internalTopicRef href="#orientador"/>
8     </instanceOf>
9   </role>
10  <role min="1" max="1">
11    <instanceOf>
12      <internalTopicRef href="#orientando"/>

```

```

13 |         </instanceOf>
14 |     </role>
15 | </association>

```

Os atributos @min e @max do elemento `role` definem a cardinalidade de cada papel de actuação. Em toda associação do tipo *orientar*, cada papel será encontrado uma única vez.

### B.3.2 Condições estruturais não-rígidas

As restrições apresentadas na subsecção anterior são rígidas na estruturação de tópicos ou associações. Porém, em determinadas situações é necessário deixar em aberto certas condições e validar outras.

Seguindo o exemplo apresentado anteriormente, a definição de uma regra XTche onde as associações do tipo *orientar* devem ter os papéis de actuação *orientador* e *orientando*, permitindo ainda que outros papéis sejam agregados a esta associação, encontra-se na Figura B.2.

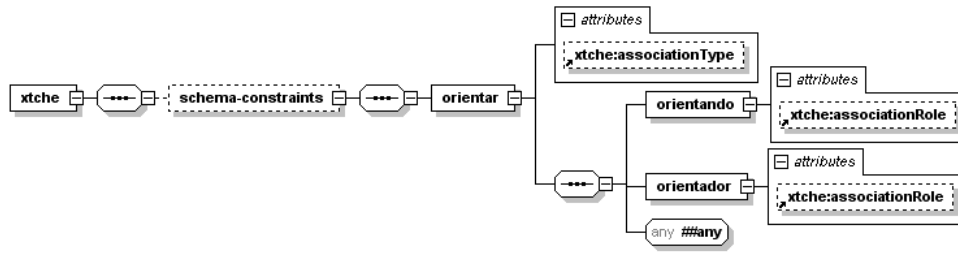


Figura B.2: Regra XTche para a associação *orientar* referente a um *open maplet pattern*

O diferencial desta especificação, em relação a especificação da Figura B.1, é o elemento “any”: ele define que qualquer papel de actuação pode ser encontrado além das regras impostas nessa restrição de esquema; portanto, desde que os papéis definidos (*orientador* e *orientando*) estejam presentes, a associação será considerada válida.

Em AsTMa!, usa-se um *maplet pattern* aberto para a definição deste tipo de restrição, conforme apresentado a seguir:

```

1 | forall $a [ (orientar) ]
2 |     => exists $a [ (orientar)
3 |                     orientador : *
4 |                     orientando : * ]

```

Esta definição está de acordo com a associação apresentada na subsecção anterior, pois o tipo e os papéis de actuação da associação estão de acordo com o *maplet pattern*. As características extras (neste caso, o papel *universidade* sendo desempenhado pelo tópico *UMinho*) não afectam a combinação, pois o *maplet pattern* em questão é aberto.

No caso de restrições estruturais não-rígidas, OSL não possui mecanismos que sejam capazes de validar tal tipo de restrição.



### B.3.3 Restrições de Existência

Este tipo de restrição determina que deve existir ao menos um tópico ou uma associação dentro do topic map que esteja de acordo com ela, para torná-lo válido. Por exemplo: para ser válido, o topic map deve possuir uma associação do tipo *nacionalidade* em que determine que *Brasil é a nação de Giovanni*.

A Figura B.3 apresenta a definição, em XTche, de uma regra onde as associações do tipo *nacionalidade* devem ter o papel de actuação *cidadão* sendo desempenhado pelo tópico *Giovani* e o papel *nação* sendo desempenhado pelo tópico *Brasil*.

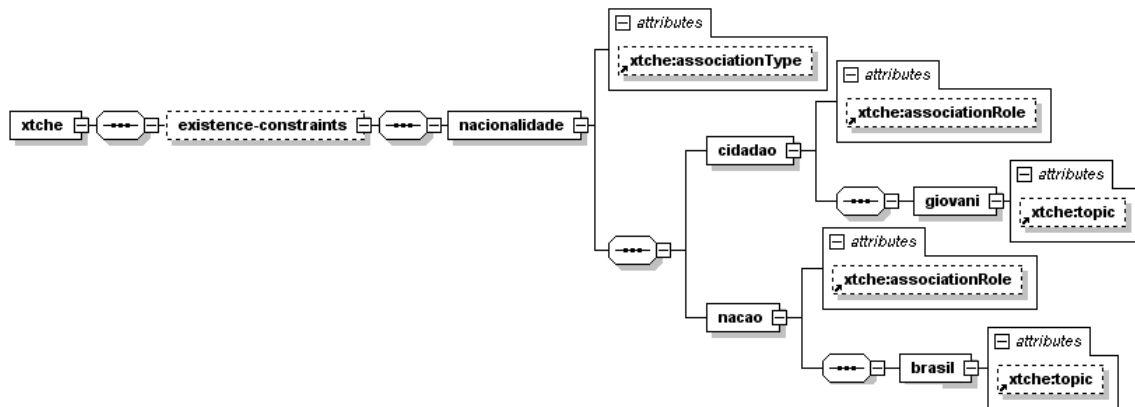


Figura B.3: Regra XTche para a verificação da existência da associação *nacionalidade*, com os papéis *cidadão* e *nação* sendo desempenhados por *Giovani* e *Brasil*, respectivamente

A restrição da Figura B.3 é uma **restrição de existência** em XTche (realizada no elemento **existence-constraints**) que determina a verificação, pelo TM-Validator, da presença de uma associação do tipo *nacionalidade*, com os papéis de actuação *nação* e *cidadão* sendo desempenhados pelos membros *Brasil* e *Giovani*. Caso não exista nenhuma associação com estas características no topic map a ser validado, o mesmo é considerado incorrecto.

Em AsTMa!, essas restrições são realizadas pela palavra-chave *exists*. Um exemplo deste tipo de restrição encontra-se a seguir:

```

1 | exists ] (nacionalidade)
2 |     cidadao : giovani
3 |     nacao   : brasil [

```

O *maplet pattern* acima é equivalente à especificação XTche da Figura B.3.

A linguagem OSL não suporta restrições de existência.

### B.3.4 Restrições Condicionais

Por vezes, as restrições devem ser aplicadas a sub-conjuntos do topic map, e não a todo o documento.

Para exemplificar, apresenta-se o topic map abaixo, que possui os tópicos *prh*, *giovani* e *gustavo*, além duma associação entre os dois primeiros.

```

1  <topic id="prh">
2    <instanceOf>
3      <topicRef xlink:href="#pessoa"/>
4    </instanceOf>
5    <baseName>
6      <baseNameString>Pedro Rangel Henriques</baseNameString>
7    </baseName>
8  </topic>
9  <topic id="giovani">
10    <instanceOf>
11      <topicRef xlink:href="#pessoa"/>
12    </instanceOf>
13    <baseName>
14      <baseNameString>Giovani Rubert Librelotto</baseNameString>
15    </baseName>
16  </topic>
17  <topic id="gustavo">
18    <instanceOf>
19      <topicRef xlink:href="#pessoa"/>
20    </instanceOf>
21    <baseName>
22      <baseNameString>Gustavo Vasconcelos Arnold</baseNameString>
23    </baseName>
24  </topic>
25  <association>
26    <instanceOf>
27      <topicRef xlink:href="#orientar"/>
28    </instanceOf>
29    <member>
30      <roleSpec>
31        <topicRef xlink:href="#orientador"/>
32      </roleSpec>
33      <topicRef xlink:href="#prh"/>
34    </member>
35    <member>
36      <roleSpec>
37        <topicRef xlink:href="#orientando"/>
38      </roleSpec>
39      <topicRef xlink:href="#giovani"/>
40    </member>
41  </association>

```

A partir do topic map acima, pretende-se definir uma regra onde todos os tópicos do tipo *pessoa* devem ser membros em associações do tipo *orientar*. Em XTche, a Figura B.4 apresenta essa restrição.

Declara-se o tópico *pessoa* com os atributos `@xtche:topicType` e `@xtche:associationPlayer`, contendo um sub-elemento *orientar* com o atributo `@xtche:associationType`; isso indica que todo tipo de tópico *pessoa* também é um membro em associações, as quais devem ser do tipo *orientar*.

Ao confrontar o topic map acima com a regra da Figura B.4, o TM-Validator detectará que o tópico *prh* a satisfaz, pois é um tópico do tipo *pessoa* e está participando de uma associação do tipo *orientar*. O mesmo acontece com o tópico *giovani*. Esta iteração é feita até que todos os sub-mapas (ou seja, os tópicos do tipo *pessoa*) encontrados tenham sido

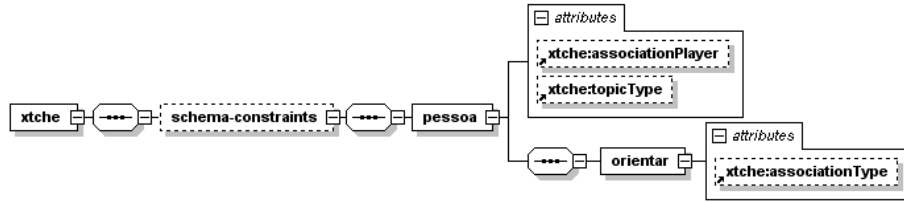


Figura B.4: Regra XTche para definir que todo t pico do tipo *pessoa* deve estar presente em uma associa  o do tipo *orientar*

verificados. Assim sendo, acusar-se-  que o t pico *gustavo*   uma inst ncia de *pessoa* e n o participa de uma associa  o do tipo *orientar*. Com isso, esse topic map n o   considerado v lido.

Essa regra   equivalente   cl usula em AsTMa! mostrada a seguir. Para exemplificar, considere o seguinte *maplet pattern*:

```

1 forall [ $p (pessoa) ]
2   => exists [ (orientar)
3     * : $p ]

```

Essa regra define que todo t pico do tipo *pessoa* deve participar de uma associa  o do tipo *orientar*, independente do papel de actua  o de cada pessoa.

A primeira cl usula *forall*   a cl usula de condi  o e actua como um selector de sub-mapas. A cl usula *exists* validar  os sub-mapas seleccionados a partir da cl usula condicional. A vari vel *\$a* limita-se ao contexto desta cl usula em quest o; assim, a vari vel *\$a* indica que a inst ncia de *pessoa* encontrada no *forall* participa como membro da associa  o *orientar* (cl usula *exists*).

### B.3.5 Restri  es Booleanas de Nega  o

Esta subsec  o discute o uso de operadores booleanos de nega  o que construir o as restri  es booleanas, pois por vezes pode ser necess rio proibir a exist ncia dum t pico espec fico (ou associa  o) dentro do topic map.

Por exemplo: em um topic map *tm*, n o pode existir nenhuma inst ncia do t pico *orientador* que possua uma ocorr ncia, seja ela qual for.

Em XTche, a defini  o dessa restri  o est  representada na Figura B.5.

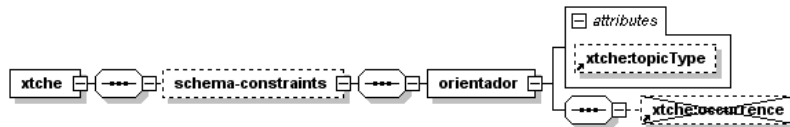


Figura B.5: Regra XTche para definir que t pico do tipo *orientador* n o pode ter nenhuma ocorr ncia

Usando a definição de cardinalidade em XML Schema, XTche permite a especificação dos objectos que não devem ser encontrados no domínio de um topic map. Assim, para determinar que todo tópico do tipo *orientador* não deve possuir ocorrências, os atributos `@minOccurs` e `@maxOccurs` devem estar com o valor “0” (zero), conforme o código abaixo:

```
1 | <xs:element ref="xtche:occurrence" minOccurs="0" maxOccurs="0"/>
```

Para essas situações, AsTMa! propicia o uso do operador booleano *not*. Este operador de negação pode somente ser utilizado por restrições de existências, isto é, podem aparecer somente em frente à palavra-chave *exists*, como mostra o seguinte exemplo:

```
1 | not exists [ * (orientador)
2 |           oc : * ]
```

### B.3.6 Restrições Booleanas – “E” e “OU”

Os operadores booleanos podem ser utilizados em casos onde, por exemplo, todo tópico ou associação será válido se estiver de acordo com a regra A ou com a regra B.

Como exemplo prático, se quer confirmar que toda associação do tipo *conter* possui um papel de actuação *contem* desempenhado por um tópico do tipo *país*, enquanto o papel *está-contido* ou é desempenhado por um tópico do tipo *estado* ou do tipo *cidade*.

XTche permite a definição desta restrição tomando como partido duas características de XML Schema: a opção *choice* para o “ou” e a opção *sequence* para o “e”.

Na Figura B.6, percebe-se que o papel de actuação *está-contido* utiliza o *choice* para afirmar que ele será desempenhado ou por um *estado* ou por uma *cidade*. Por sua vez, o tipo de associação *conter* utiliza o *sequence* para afirmar que é composto por dois papéis de actuação: *contém* e *está-contido*.

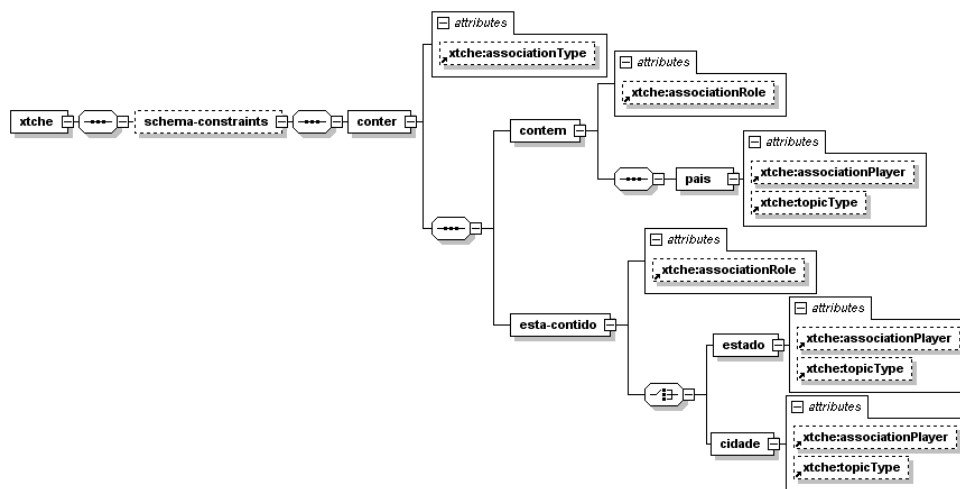


Figura B.6: Regra XTche para definir opcionalidade

A apresentação correcta dessa mesma restrição em AsTMa! está abaixo representada:

```
1 forall [ (conter)
2     contem: $a
3     esta-contido: $b
4 ] => exists [ $a (pais) ] and exists [ $b (estado) ] or exists [ $a (pais) ] and exists [ $b
5 (cidade) ]
```

Dessa forma, conclui-se que este tipo de restrições é suportado tanto por AsTMa! como por XTche.